(2

AD

US ARMY MATERIEL COMMAND

MEMORANDUM REPORT BRL-MR-3443

AD-A158 344

COMPUTER IMPLEMENTATION OF A MUZZLE BLAST PREDICTION TECHNIQUE

Charles W. Heaps
Kevin S. Fansler
Edward M. Schmidt

May 1985

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER MEMORANDUM REPORT BRL-MR-3443	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
COMPUTER IMPLEMENTATION OF A MUZZLE BLAST PREDICTION TECHNIQUE

5. TYPE OF REPORT & PERIOD COVERED
Final

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Charles W. Heaps, Kevin S. Fansler, Edward M. Schmidt

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. Army Ballistic Research Laboratory
ATTN: AMXBR-LFD
Aberdeen Proving Ground, MD 21005-5066

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
RDT&E 1L161102AH43

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Army Ballistic Research Laboratory
ATTN: AMXBR-OD-ST
Aberdeen Proving Ground, MD 21005-5066

12. REPORT DATE
May 1985

13. NUMBER OF PAGES
89

14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Gun Muzzle Blast;
Blast Wave Scaling
Overpressure
Surface Pressure

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (ner)

A computer code is developed which generates contour maps of peak incident overpressure, peak reflected overpressure, blast wave time of arrival, and blast wave positive phase duration. The code is based upon a free field muzzle blast scaling technique which is extended to treat blast wave reflection from surfaces. The predictions of the code are shown to compare reasonably well with experiment.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

High levels of muzzle blast overpressure can have adverse effects on weapon crew members, nearby structures, and instrumentation. Control of these effects requires the ability to predict the details of the blast pulse as a function of weapon design and emplacement, propellant and projectile characteristics, and launch conditions. Additionally, techniques to reduce or control the blast through changes in these properties are of practical importance. Fansler and Schmidt[1] developed scaling relations that permit estimation of peak incident overpressure, blast wave time of arrival, and positive phase duration. Comparison of these estimates with experiment demonstrates that the scaling approach provides a reasonable initial estimate of muzzle blast characteristics. Since the relations are sensitive to weapon launch conditions and propellant charge design, they may be used to study the influence of these properties on muzzle blast.

The present work has two main objectives. The first is to extend the scaling approach to treat the problem of pressure loadings on surfaces adjacent to the weapon. The second is to use the scaling relations in the development of a computer code called BLAST that plots contour maps of the muzzle blast quantities. We believe that the results obtained from the scaling relations are most easily interpreted when presented in this form.

The remainder of this report may be outlined as follows. First we present the scaling relations used to calculate the incident overpressure, blast wave time of arrival, and positive phase duration at points on a surface that can have any desired orientation with respect to the cannon boreline. From the calculations of incident overpressure and time of arrival, the reflected blast overpressure on the surface of interest is determined. For all of the blast properties, an algorithm is established to generate contour plots. Representative plots obtained from BLAST are selected for comparison with experimental results.

II. FREE FIELD MUZZLE BLAST CALCULATIONS

The free field blast computations are performed using scaling relations developed by Fansler and Schmidt. Their scaling approach has been described in detail;[1] hence, only the final results of the work are presented here. The quantities of interest in the free field blast problem are the peak incident overpressure in atmospheres, $\bar{P}$, the blast wave time of arrival, t_a, and the positive phase duration, τ. The expressions derived for these quantities are summarized by

$$\bar{P} = 2.4 Z \tag{1a}$$

1. *K. S. Fansler and E. M. Schmidt, "The Prediction of Gun Muzzle Blast Properties Utilizing Scaling," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, BRL Technical Report ARBRL-TR-02504, July 1983 (AD B 075859L).*

$$t_a = \frac{r}{a_\infty} f(Z) - \frac{a_\infty}{\ell'}(0.94 \cos\theta + 9.24) \qquad (1b)$$

$$\tau = (\ell'/a_\infty)\,[1 + 0.13\,(r/\ell')] \qquad (1c)$$

where

$$\ell' = \ell\,[\mu\cos\theta + (1 - \mu^2\sin^2\theta)^{1/2}] \qquad (1d)$$

$$Z = (r/\ell')^{-1.1} \text{ , and} \qquad (1e)$$

$$f(Z) = 1 + 10Z - (Z^2/1.2) + (Z^3/2.3) - (Z^4/3.4) + (Z^5/4.5) - (Z^6/5.6). \qquad (1f)$$

For subsonic exit flow ($V_p \leq a_m$)

$$\ell = D\left\{\frac{(8.62 \times 10^{-3})\, p_m a_m}{(\gamma-1)\, p_\infty a_\infty}\left[1 + \frac{\gamma(\gamma-1)}{2}\right]\left[\frac{2}{\gamma+1}\left(1 + \frac{(\gamma-1)}{2}\frac{V_p}{a_m}\right)\right]^{\frac{3\gamma-1}{\gamma-1}}\right\}^{1/2}. \qquad (2)$$

For supersonic exit flow ($V_p > a_m$),

$$\ell = (9.28 \times 10^{-2})\, D\left[\frac{p_m V_p}{(\gamma-1)\, p_\infty a_\infty}\left(1 + \frac{\gamma(\gamma-1)\, V_p^2}{2a_m^2}\right)\right]^{1/2}. \qquad (3)$$

To use these relations, it is necessary to know the weapon characteristics and propellant gas properties at shot ejection. When the

latter are not known from experiment, they may be determined using interior ballistics theory. The user of BLAST has the option of employing the following simplistic interior ballistics model to obtain the gas properties at shot ejection.

Lagrangian interior ballistics as given by Corner[2] is utilized. The internal energy of the propellant gases immediately prior to projectile ejection is given by

$$E = \frac{BCRT_{mean}}{\gamma-1} = \frac{BCRT_a}{\gamma-1} - \tfrac{1}{2}(m_1 + C/3)\ (1 + \chi)\ V_p^{\,2} \tag{4}$$

where B is the fraction of propellant burnt, C is the propellant mass, m_1 is the effective projectile mass accounting for bore friction, and χ is the ratio of heat losses to kinetic energy. The following semi-empirical formulation for χ is used

$$\chi = \frac{(10500)\ UD^{3/2}\ (T_a - 300)\ \Omega}{A_e\ V_p^{\,2}\ (m_p + C/3)\ [1.7 + 671D^{1/2}\ (D^2/C)^{.86}]} \tag{5}$$

where Ω is the roughness factor. In this report the roughness factor is taken as 1.35, an average of large and small gun values. If we assume an ideal gas the muzzle pressure is given by

$$p_m = \frac{(\gamma-1)\ E}{U\ [1 + C/(3m_1)]} \tag{6}$$

where U is the combined volume of the bore and chamber. The following expression is obtained for the exit sound speed of the propellant gases

$$a_m = (\gamma RT_m)^{1/2} = \left\{\gamma(\gamma-1)\ E/[C + C^2/(3m_1)]\right\}^{1/2} \quad . \tag{7}$$

Results obtained from this simple theory were compared with those of a more exact computer model[3] and shown to produce similar results. Once the weapon

---

2. *J. Corner, Theory of the Interior Ballistics of Guns, John Wiley, New York, 1950.*

3. *Private communication from Brian Bertrand.*

characteristics and launch conditions are determined, the scaling relations given in Equation (1) are used to compute $\overline{P}$, t_a, and τ as functions of r and θ. The contour plotting algorithm requires the evaluation of these functions at regularly spaced points (gridpoints) in the plotting domain. The values of r and θ must therefore be determined for each gridpoint. The geometry of the problem is shown in Figure 1. The origin of the polar coordinate system in which contours are plotted is the perpendicular projection of the muzzle of the gun into the plane of interest. The distance from the origin to the muzzle is h, and ϕ is the angle between the plane and the boreline of the gun. The contour algorithm requires a rectangular grid in the contour plane. For this reason, the gridpoints are located in a cartesian coordinate system originating at the muzzle of the gun. The x-y plane of this coordinate system is taken parallel to the contour plane; thus, any point in the contour plane is defined by (x,y,-h). The distance from the muzzle to a point in the plane is given by the magnitude of $\vec{r}$:

$$r = (x^2 + y^2 + h^2)^{1/2}. \tag{8}$$

Let $\vec{u}$ be a unit vector parallel to the boreline. The angle θ between $\vec{u}$ and $\vec{r}$ is given by

$$\theta = \cos^{-1} \frac{\vec{r} \cdot \vec{u}}{r} = \cos^{-1} \left[\frac{y\cos\phi - h\sin\phi}{(x^2 + y^2 + h^2)^{1/2}} \right] . \tag{9}$$

The values of r and θ may be used to obtain $\overline{P}$, t_a, and τ from the scaling relations at each point in the contour plane.

III. REFLECTED SHOCK WAVE CALCULATIONS

The reflection of shock waves from surfaces can be quite complex. Edney[4] has identified a number of possible flow structures depending upon the strength of the incident wave and the angle of reflection. In the present context, the process is considerably simplified. Only two types of reflection are considered: regular and single Mach stem.[5]

---

4. *B. E. Edney, Effects of Shock Impingement on the Heat Transfer Around Blunt Bodies," AIAA Journal, Vol. 16, No. 1, January 1968, pp. 15-21.*

5. *B. P. Bertrand, "Measurement of Pressure in Mach Reflection of Strong Shock Waves in a Shock Tube," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland, BRL-MR-2196, June 1972 (AD 746613).*

For points very close to the reflecting surface (the contour plane), the blast may be considered planar. The shock wave angle of incidence, α_1, is simply the angle between the direction of propagation of the blast and the surface normal, $\vec{n}$, to the contour plane (Figure 2). One method of obtaining the direction of propagation of the blast wave is to evaluate the gradient of t_a since the shape of the wave is equivalent to the surface defined by constant t_a. This gradient is normal to the blast wave surface; therefore, it points in the direction of propagation. The difficulty with this approach is that the computer time required to evaluate the gradient of t_a at each grid point is excessive. In Appendix A we describe an alternate method for calculating the shock wave angle of incidence. This alternate method may be shown to be mathematically equivalent to the gradient approach, but it is more efficient computationally. In BLAST we use the approach described in Appendix A.

Once the shock wave angle of incidence, α_1, is obtained, we determine whether regular reflection or Mach reflection of the shock wave occurs. The first step in this procedure is to shift from a fixed reference frame to one moving along the reflecting surface at the same velocity as the shock waves. In this reference frame the shock wave is stationary and has a streamline flowing through it parallel to the reflecting surface (Figure 3). The relations for compressible flow through an oblique shock wave[6] are applicable to this system.

The Mach number of the streamline in the region in front of the incident shock is given by

$$M_1 = \left\{\left[\frac{\gamma+1}{2\gamma}\left(\frac{p_I-p_\infty}{p_\infty}\right) + 1\right]/\sin^2\alpha_1\right\}^{1/2} \tag{10}$$

where p_I and p_∞ are the pressures behind and in front of the incident shock, respectively. Across the incident shock, the flow is deflected through an angle, δ_1, given by

$$\delta_1 = \tan^{-1}\left[2(\cot\alpha_1)\,\frac{M_1^2\sin^2\alpha_1-1}{M_1^2(\gamma+\cos 2\alpha_1)+2}\right] \tag{11}$$

6. *H. W. Liepmann and A. Roshko, Elements of Gasdynamics, J. Wiley, New York, 1957.*

and the Mach number behind the incident shock is obtained from

$$M_2 = \left\{ \frac{(\gamma-1)\,\frac{p_I}{p_\infty} + (\gamma-1)}{2\gamma\,\frac{p_I}{p_\infty}} \right\}^{1/2} \Big/ \sin(\alpha_1 - \delta_1)\ . \tag{12}$$

The question of whether regular reflection is possible may now be addressed. The maximum stream deflection for a specified upstream Mach number is given approximately by

$$\delta_{max} = \frac{4}{3\sqrt{3}\,(\gamma+1)}\,\frac{(M_1^{\,2} - 1)^{3/2}}{M_1^2}\ . \tag{13}$$

If $\delta_1 \geq \delta_{max}$, regular reflection is not possible and Mach reflection occurs. If $\delta_1 < \delta_{max}$, regular reflection occurs and the analytic solution for the reflected overpressure may proceed.

If regular reflection is possible, the flow is deflected through the reflected shock. This deflection angle, δ_2, must be equal to δ_1 because the boundary condition requires the flow to be parallel to the reflecting surface. Equation (11) applied to the flow across the reflected shock, with $\delta_1 = \delta_2$, yields

$$2\,(\cot \alpha_2)\,\frac{M_2 \sin^2 \alpha_2 - 1}{M_2^2\,(\gamma + \cos 2\alpha_2)} - \tan \delta_2 = 0 \tag{14}$$

where α_2 is the wave angle of the reflected shock. Equation (14) is solved for α_2 by iteration. Once α_2 is known, the following equation is used to determine the pressure behind the reflected shock, p_R

$$\frac{p_R}{p_I} = \frac{2\gamma}{\gamma+1}\,(M_2^{\,2} \sin^2 \alpha_2 - 1) + 1\ . \tag{15}$$

Finally, the reflected overpressure in atmospheres is given by

$$\bar{P}_R = \frac{p_R - p_\infty}{p_\infty} = \frac{p_R}{p_I}\frac{p_I}{p_\infty} - 1 \quad . \tag{16}$$

A computational problem arises in the preceding development as α_1 approaches zero. The expression for M_1 given in Equation (10) becomes arbitrarily large due to the sin α_1 term in the denominator. To avoid this difficulty we do not use the relations presented above for angles of incidence less than one degree. For angles of incidence less than this value, the solution for plane shock wave reflection normal to the surface is adequate. In this special case the ratio of reflected overpressure to incident overpressure is given by

$$\frac{p_R}{p_I} = \frac{(3\gamma-1)\ p_I - (\gamma-1)}{(\gamma-1)\ p_I + (\gamma+1)} \quad . \tag{17}$$

If regular reflection does not occur ($\delta_1 \geq \delta_{max}$) it is impossible to obtain an analytic solution for the reflected overpressure using the theory of simple oblique waves. In this case the reflected overpressure is extracted from empirical results. A typical data plot of reflected overpressure versus shock wave angle of incidence for various incident overpressures[7] is presented in Figure 4. Similar data from a variety of sources were examined and two general features noted. First, beyond a certain angle of incidence whose value depends on the incident overpressure, the reflected overpressure curves may be approximated by straight lines. To determine the equations of these lines a point and slope are needed. At $\alpha_1 = 90^0$, the blast wave is at grazing incidence to the surface and the ratio of reflected to incident overpressure equals one, independent of the incident overpressure. Thus, the right end point of all the straight lines is defined. The slopes of the lines are taken to be a function of the incident overpressure and are obtained for 11 values of this parameter. Lagrangian interpolation is then used to obtain a slope for any intermediate value of incident overpressure. The interpolation procedure is also used to determine the angle of incidence beyond which the linear approximation is valid.

---

7. *S. Glasstone, Editor, The Effects of Nuclear Weapons, Dept. of the Army Pamphlet No. 50-3, March 1977, p. 123.*

The second feature apparent in the empirical plots pertains to the region near the maxima of the reflected overpressure curves. For values of α_1 between the regular reflection and the linear interpolation regimes, the curves have a shape that can be approximated by cubic polynomials in α_1. The four boundary values needed to evaluate the polynomial are provided from the solutions at points A and B of Figure 5. The reflected overpressure at point A, $\pi(\alpha_A)$, is determined from the analytic solution for regular reflection. The slope at A, $\pi'(\alpha_A)$, can be approximated by taking a finite difference. At point B, $\pi(\alpha_B)$ and $\pi'(\alpha_B)$ are obtained from the Lagrangian interpolation procedure described above. The four-term polynomial at A is

$$\pi(\alpha_A) = \pi_0 + \pi_1\alpha_A + \pi_2\alpha_A^2 + \pi_3\alpha_A^3. \tag{18}$$

The derivative of this polynomial is

$$\pi'(\alpha_A) = \pi_1 + 2\pi_2\alpha_A + 3\pi_3\alpha_A^2 \tag{19}$$

with two corresponding equations at point B. These four equations are solved for the four unknown coefficients π_0, π_1, π_2, and π_3.

The joining of the regular reflection solution, the cubic fit, and the linear fit produces a continuous variation of reflected overpressure coefficient versus angle of incidence for any specified incident overpressure. Results of this procedure are plotted in Figure 6. While this method of determining the reflected overpressure is relatively simple, it substantially reproduces the empirical plots as is demonstrated by comparison with experimental data (Figure 7).

IV. THE CONTOUR PLOTTING METHOD

The results of the analysis have been formulated as a computer code, BLAST, which is programmed in FORTRAN 77 and is designed for interactive use on graphics terminals. A listing of BLAST may be found in Appendix B.

The method used to plot the contours is relatively standard and originated from an ALGOL computer code.[8] A grid is constructed covering the

8. *B. R. Heap, et al, "Three Contouring Algorithms," NPL-81, National Physical Laboratory, Teddington, England, December 1969.*

region of space that is of interest. This grid is composed of many small rectangular cells. At each corner of every cell (a gridpoint), the value of the function to be plotted must be known. Each contour value is processed individually, i.e., all contours at a particular level are plotted before moving to the next level. The search for contours at a particular level begins by locating all the contours that intersect the outside boundary of the grid (open contours). The search along the boundary progresses counterclockwise from the bottom left corner of the grid until two adjacent gridpoints are found that bound the contour level. To avoid repeatedly finding and plotting the same contours, an additional requirement is imposed: the larger functional values must be to the right of the intersection point of the contour and the grid boundary for the point to be recognized as the beginning of a contour. Once the beginning of a contour is found, it is followed cell by cell through the grid by comparison of adjacent grid points. The precise path of the contour across any cell boundary is determined by linear interpolation. The coordinates of this interpolated point are passed to a plotting subroutine which uses a commercial graphics package, DISSPLA, to draw each line segment on the plotting device. Because the grid squares are very small, there is no perception of the contour lines being composed of straight segments.

After all the open contours at a particular level are located and plotted, the grid is searched for closed contours at that level. The closed contours, once found, are followed through the grid and plotted in the same manner as the open contours. When plotting of the closed contours is complete, the process is repeated for the next contour level.

V. RESULTS

In Figures 8 and 9 we present contour maps, generated by BLAST, of peak incident overpressure, peak reflected overpressure, blast wave time of arrival, and positive phase duration. Maps of incident overpressure, time of arrival, and positive phase duration are generated within thirty seconds on a VAX 11/780. Plots of reflected overpressure require additional computations and are produced in approximately five minutes. The predictions shown are for the 30 mm, XM230 Chain Gun. The weapon is positioned parallel to the contour plane with the muzzle 0.26 meter distant from the origin of the contour grid. The strong directional dependence of the muzzle blast field is readily apparent in the plots of incident and reflected overpressure. The contours of time of arrival represent the blast wave surface as it propagates outward from the muzzle. This surface rapidly becomes spherical with a center displaced in the forward direction. The positive phase duration map shows the rate of increase of the blast wavelength to be larger in the forward direction than to the rear of the muzzle.

To check the validity of our model, we compare BLAST predictions with experimental blast measurements obtained in a previous test program[9] of the 30mm

---

9. E. M. Schmidt, "Muzzle Blast Pressure Loadings on Aircraft Surfaces," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland, ARBRL-MR-03336, February 1984 (AD A 139132).

Chain Gun. For these tests, pressure transducers were mounted in a linear array on an aluminum plate. The plate was then aligned at angles of zero and minus five degrees relative to the line of fire, with separations of 0.26 meter and 0.23 meter respectively. For the zero degree orientation, data were collected with the transducer array aligned both parallel and normal to the line of fire. For the minus five degree orientation, only the parallel array alignment was used. In Figures 10, 11, and 12 we compare the experimental measurements of reflected overpressure, time of arrival, and positive phase duration for each of the three transducer orientations. In these plots the abscissa, S, denotes distance on the plate surface measured from the point on the plate nearest the muzzle. Negative values of S indicate positions behind the muzzle.

Figure 10 shows that the peak reflected overpressure measurements agree very favorably with BLAST predictions, particularly along the line of fire. The small plateau which is apparent in each of the predicted curves denotes the transition from regular reflection to Mach reflection of the incident wave. For this blast field, the effect appears too small to be defined experimentally. From the plots of time of arrival in Figure 11, we see that the location of minimum t_a predicted by BLAST is shifted forward from the location measured experimentally; however, the shape of the experimental t_a distribution is predicted quite well. It is obvious from Figure 12 that there is a definite deficiency in the positive phase duration scaling relation. We note that the expression for τ was determined by fitting scaled experimental data and that the scatter in this data justified only a simple linear curve fit. Clearly, a more complete analysis is desirable.

VI. CONCLUSIONS

The muzzle blast pressure distribution on a surface located in the vicinity of a cannon has been calculated. The computational procedure accounts for shock reflection processes and is based upon previously developed scaling relations describing muzzle blast. The approach is formalized in a computer program, BLAST, that calculates and plots contour maps of peak incident overpressure, peak reflected overpressure, blast wave time of arrival, and positive phase duration. The contours are plotted in a plane having an arbitrary orientation with respect to the gun tube. The overpressure and time of arrival predictions compare favorably with experimental measurements. The positive phase duration is over-predicted by BLAST due to problems in the scaling expression used. BLAST is completely interactive and requires only minutes of computer (VAX 11/780) time to run.

ACKNOWLEDGMENT

The authors would like to thank David H. Lyon for developing a number of original programming improvements used in this effort.

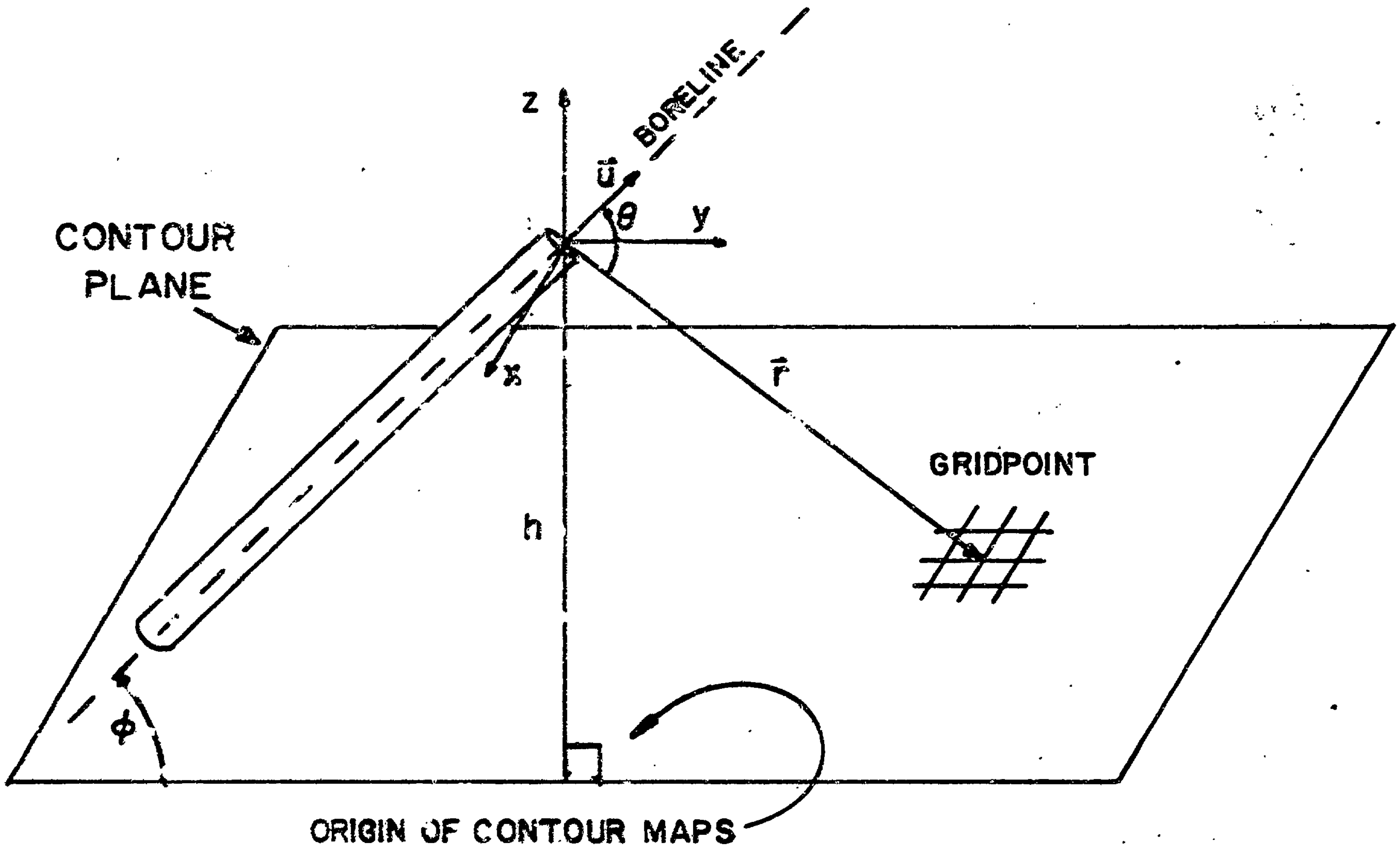


Figure 1. Geometry of the Problem of Determining r and θ at a Point in a Plane with Arbitrary Orientation with Respect to the Boreline

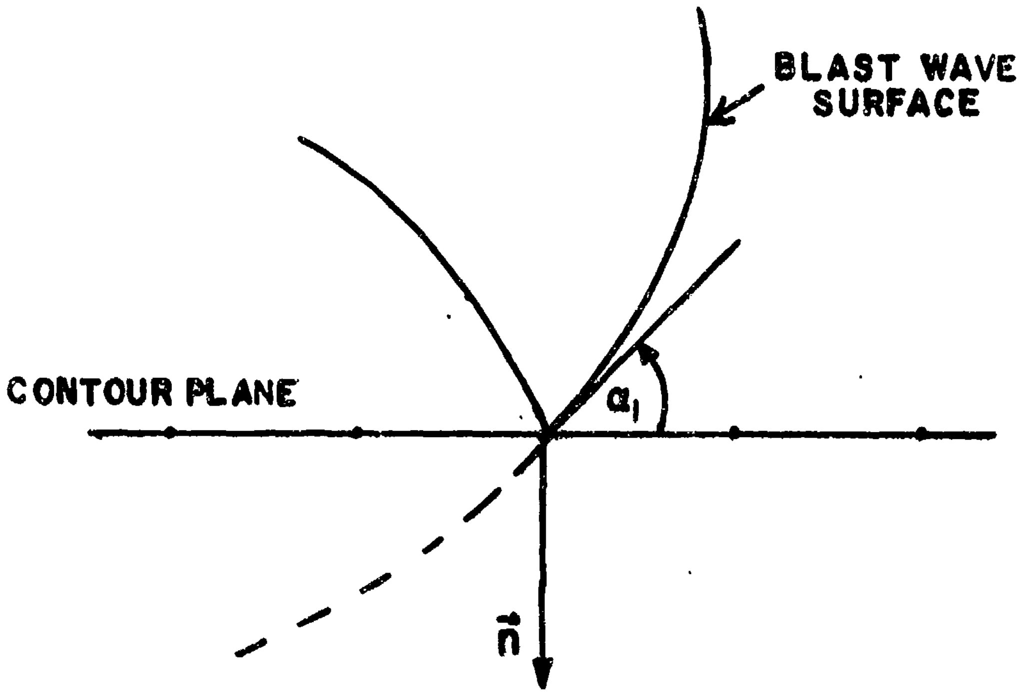


Figure 2. Geometry Illustrating the Shock Wave Angle of Incidence at a Gridpoint

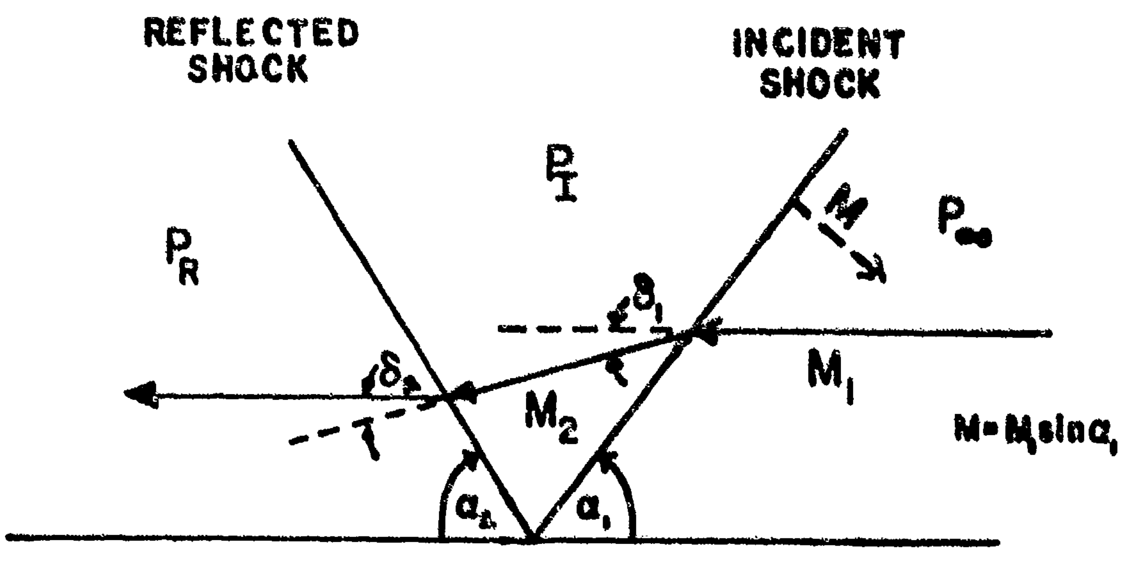


Figure 3. The Oblique Shock Wave Solution for Reflected Shock

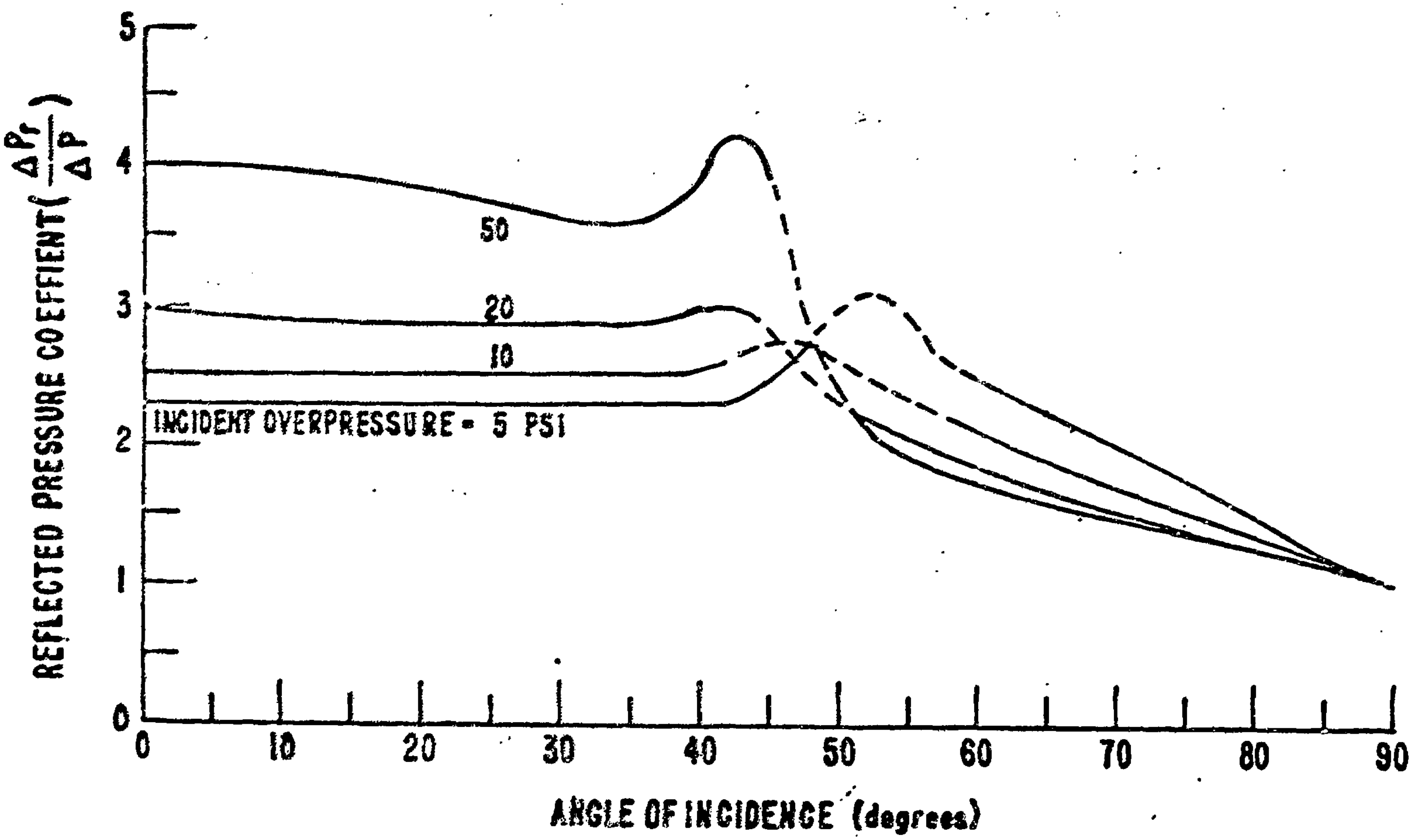


Figure 4. Reflected Pressure Coefficient vs. Angle of Incidence for Various Incident Overpressures.

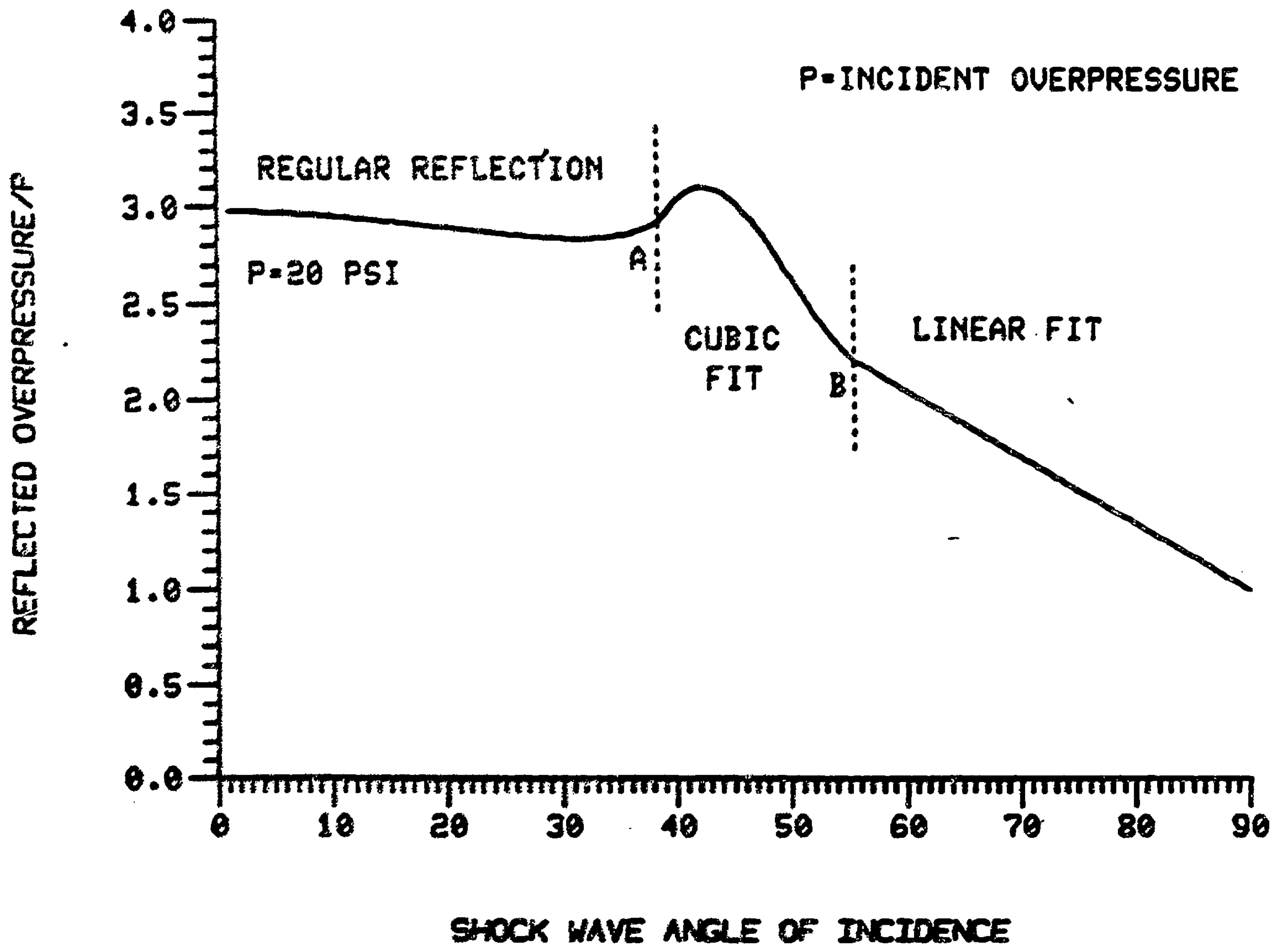


Figure 5. The Three Regions of a Typical Reflected Overpressure vs. Angle of Incidence Curve (BLAST Generated)

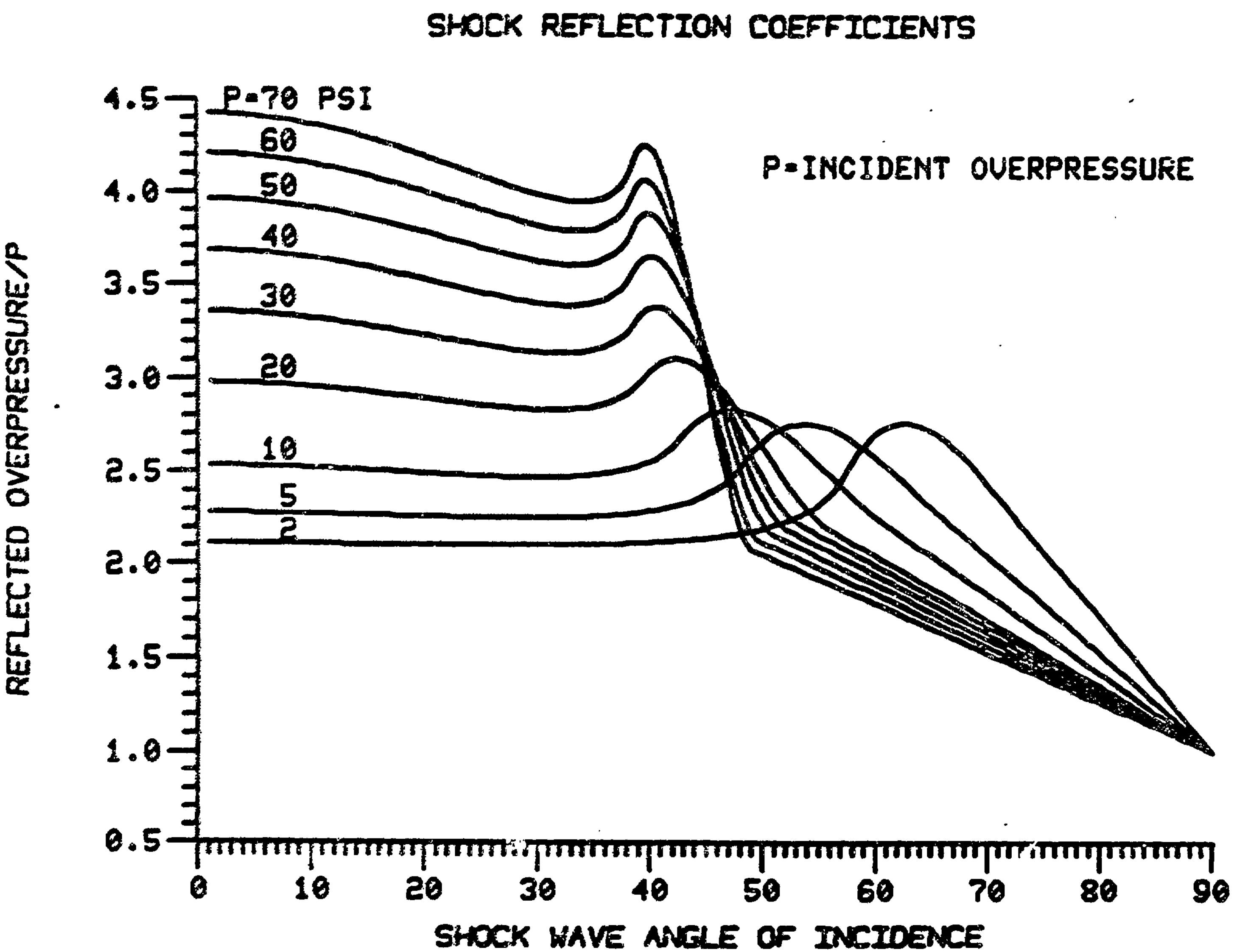


Figure 6. Reflected Pressure Coefficient vs. Angle of Incidence for Various Incident Overpressures, Obtained by Method used in BLAST

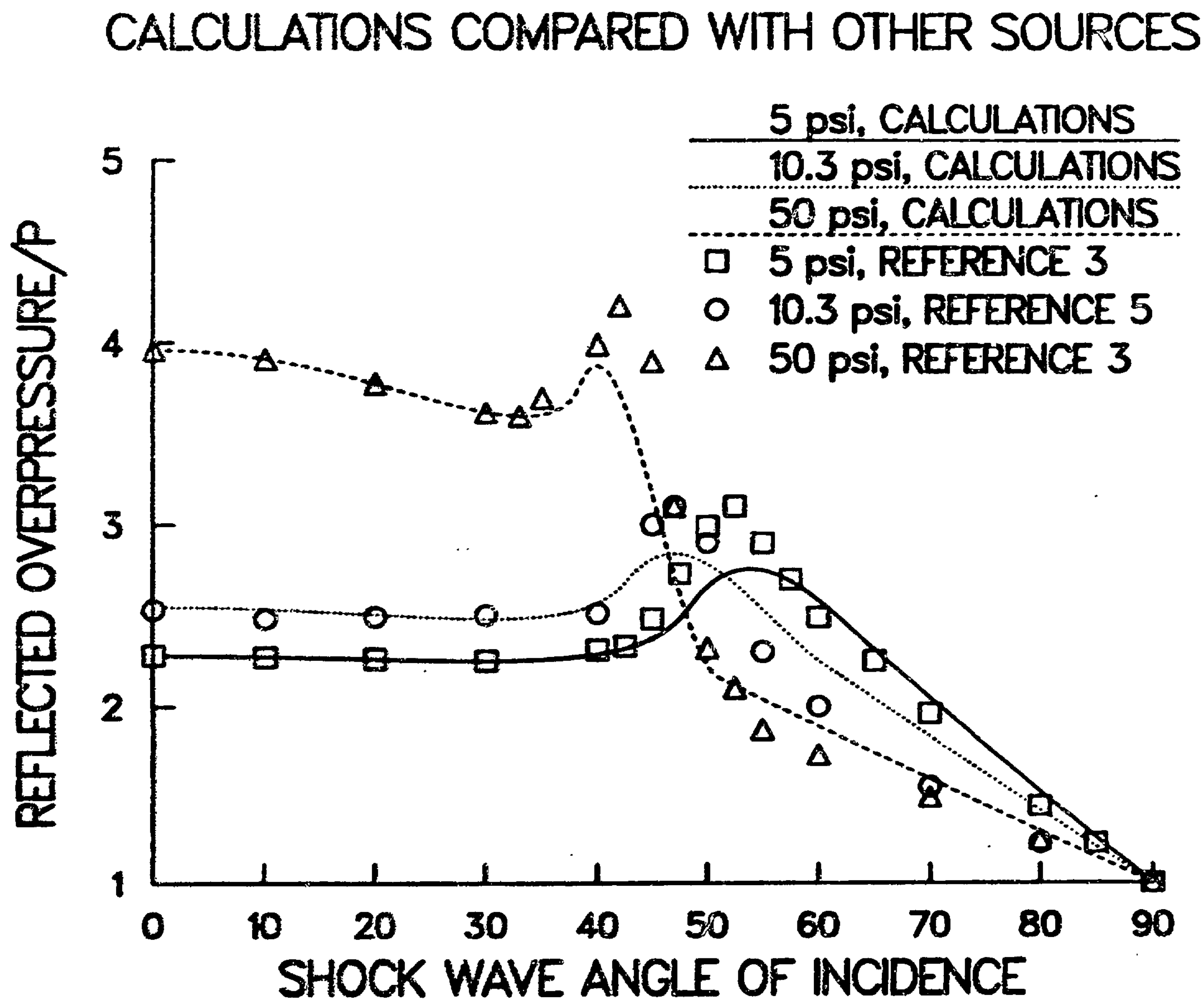
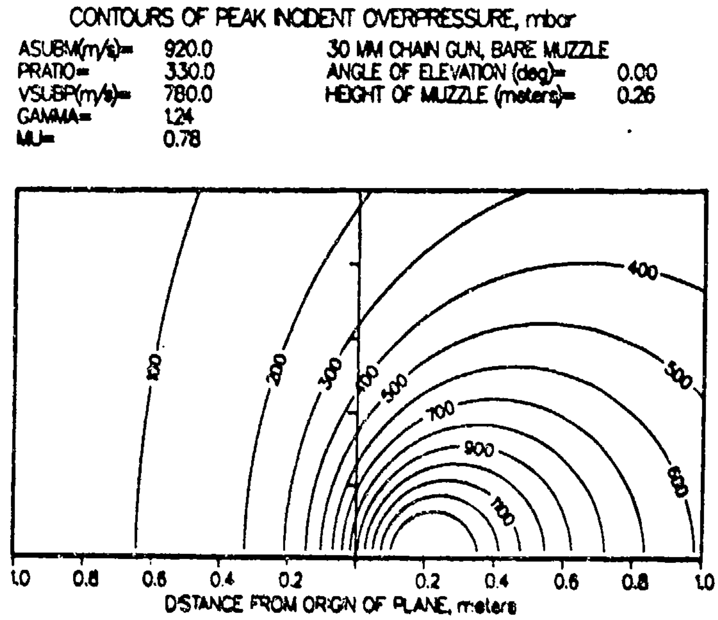


Figure 7. Comparisons Between Reflection Pressure Results Obtained with BLAST and from Other Sources

a.



b.

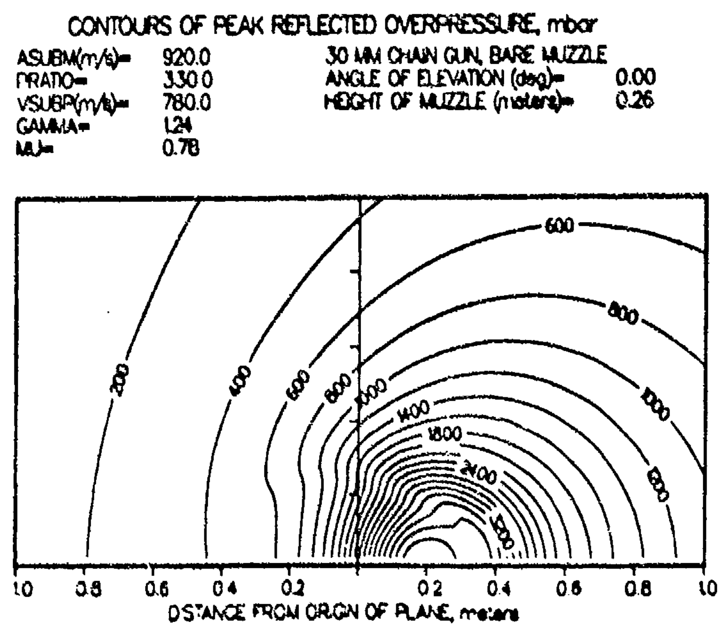
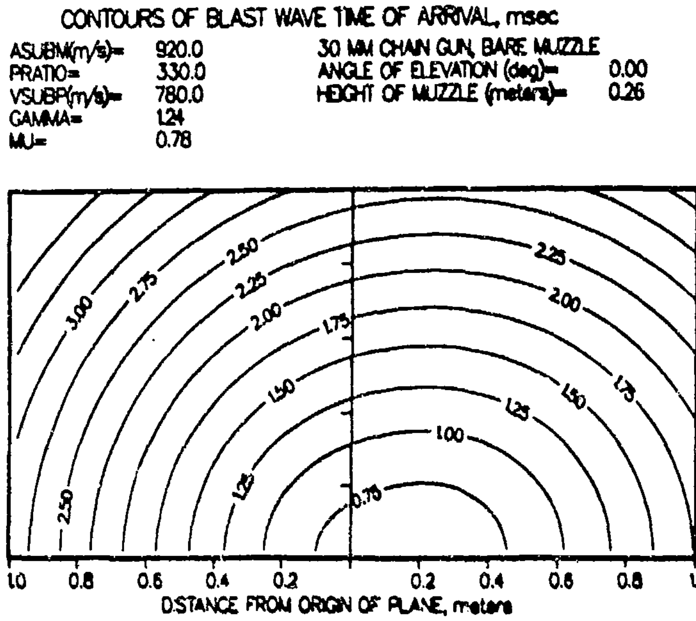


Figure 8. Contour Maps of Peak Incident Overpressure (a) and Peak Reflected Overpressure (b)

a.



b.

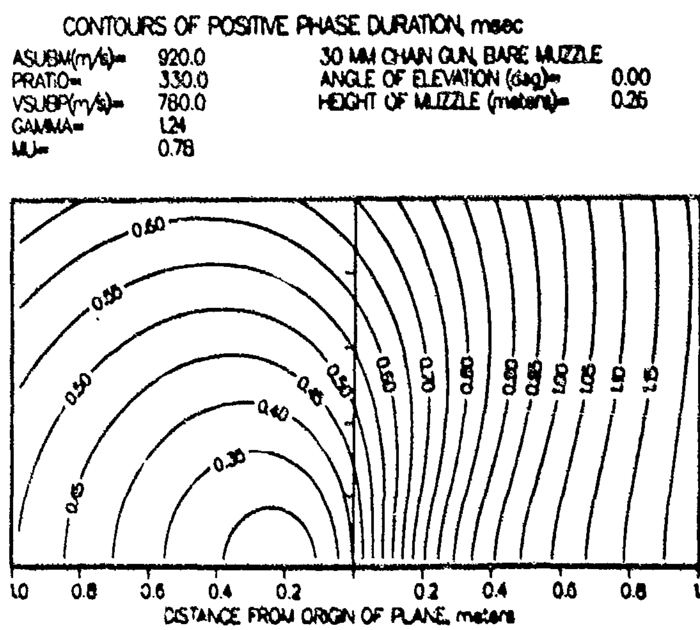


Figure 9. Contour Maps of Blast Wave Time of Arrival (a) and Positive Phase Duration (b)

a. 0° Plate Orientation, Parallel to Line of Fire

b. 0° Plate Orientation, Perpendicular to Line of Fire

c. -5° Plate Orientation, Parallel to Line of Fire

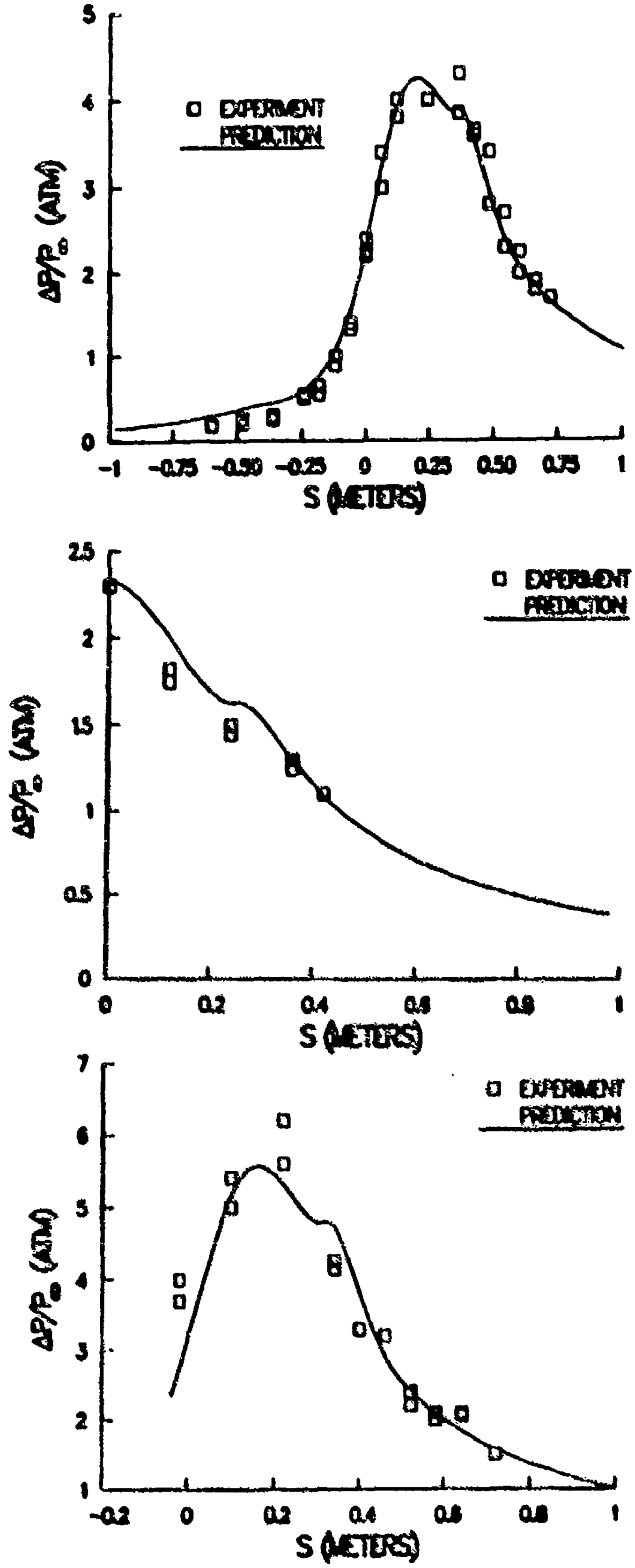
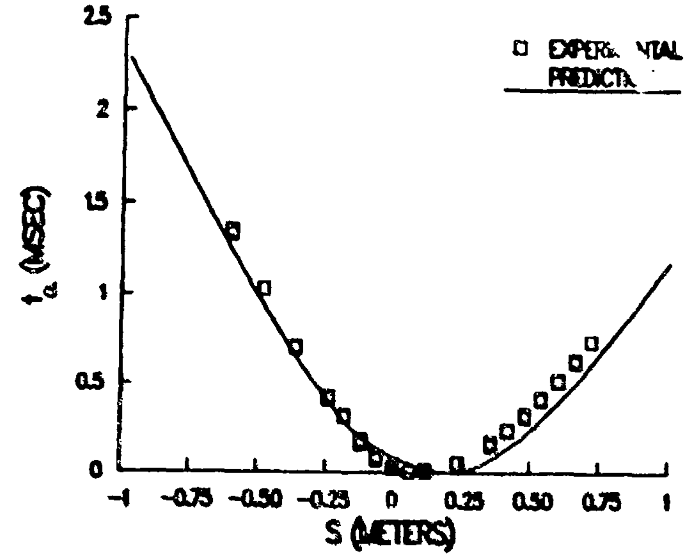
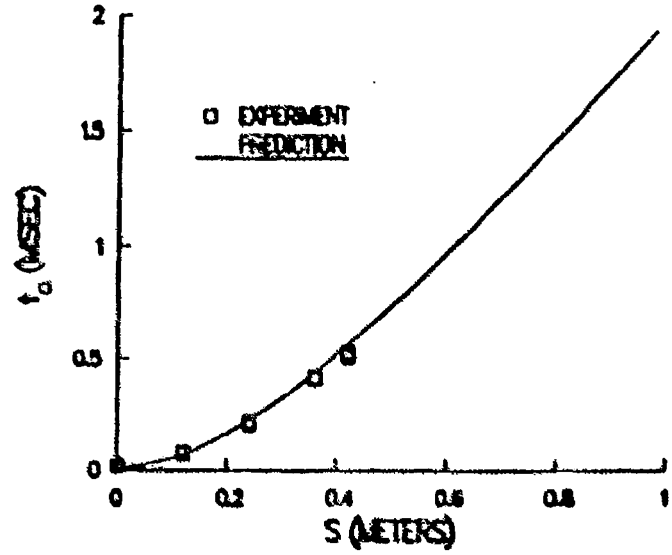


Figure 10. Comparison of Predicted Peak Reflected Overpressure with Experiment

a. 0° Plate Orientation, Parallel to Line of Fire



b. 0° Plate Orientation, Perpendicular to Line of Fire



c. -5° Plate Orientation, Parallel to Line of Fire

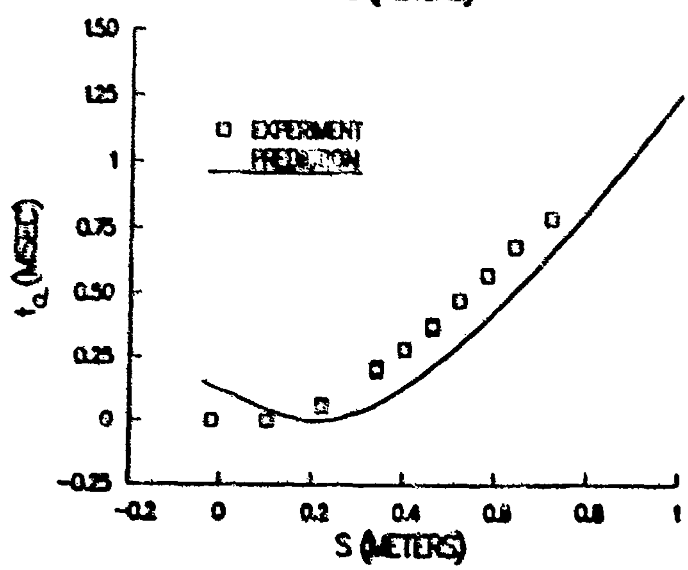
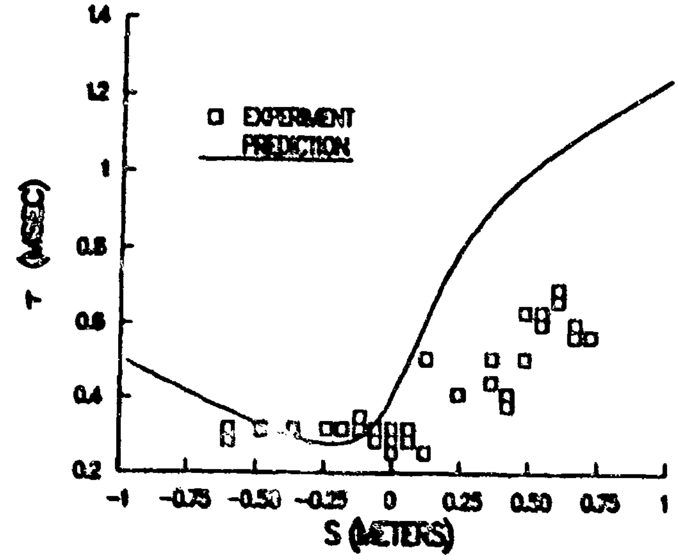
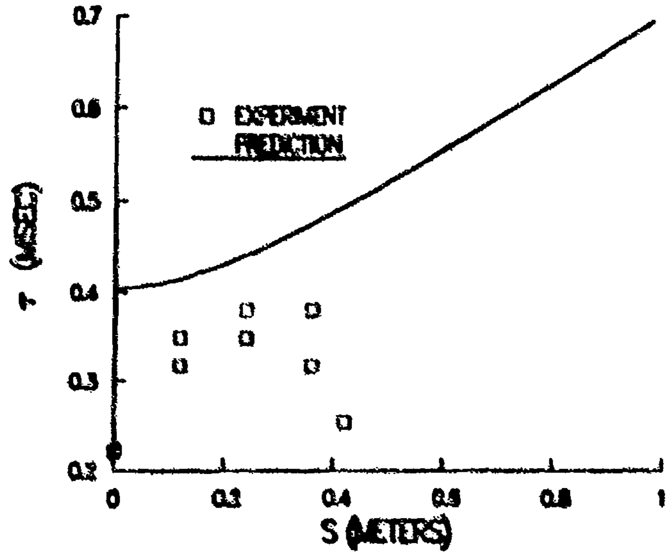


Figure 11. Comparison of Predicted Blast Wave Time of Arrival with Experiment

a. 0° Plate Orientation, Parallel to Line of Fire



b. 0° Plate Orientation, Perpendicular to Line of Fire



c. -5° Plate Orientation, Parallel to Line of Fire

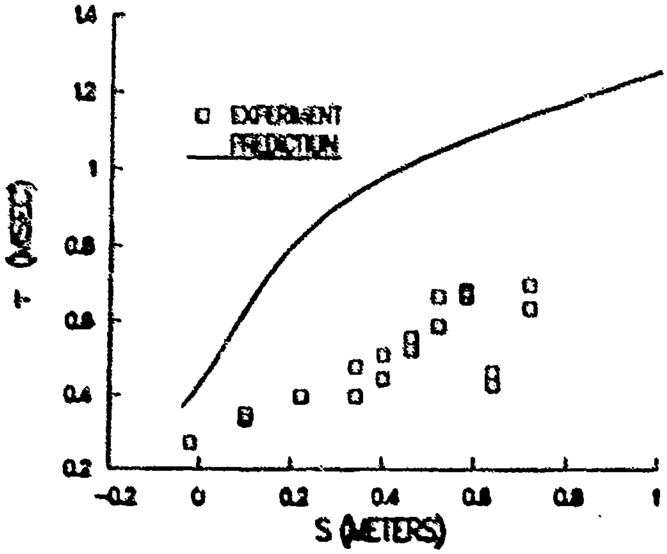


Figure 12. Comparison of Predicted Positive Phase Duration with Experiment

REFERENCES

1. K. S. Fansler and E. M. Schmidt, "The Prediction of Gun Muzzle Blast Properties Utilizing Scaling," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, BRL Technical Report ARBRL-TR-02504, July 1983 (AD B075859L).

2. J. Corner, Theory of the Interior Ballistics of Guns, John Wiley, New York, 1950.

3. Private communication from Brian Bertrand.

4. B. E. Edney, Effects of Shock Impingement on the Heat Transfer Around Blunt Bodies," AIAA Journal, Vol. 16, No. 1, January 1968, pp. 15-21.

5. B. P. Bertrand, "Measurement of Pressure in Mach Reflection of Strong Shock Waves in a Shock Tube," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland, BRL-MR-2196, June 1972 (AD 746613).

6. H. W. Liepmann and A. Roshko, Elements of Gasdynamics, J. Wiley, New York, 1957.

7. S. Glasstone, Editor, The Effects of Nuclear Weapons, Dept. of the Army Pamphlet No. 50-3, March 1977, p. 123.

8. B. R. Heap, et al, "Three Contouring Algorithms," NPL-81, National Physical Laboratory, Teddington, England, Dec 1969.

9. E. M. Schmidt, "Muzzle Blast Pressure Loadings on Aircraft Surfaces," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland, ARBRL-MR-03336, February 1984 (AD A 139132).

APPENDIX A

Analysis for Obtaining the Shock Wave Angle of Incidence

We wish to obtain the angle between the incident shock and the contour plane. In Figure A-1, A is the point of interest in the contour plane, $\vec{r}$ is the vector directed from the muzzle to A, θ is the angle between the boreline and $\vec{r}$, ϕ is the angle between the boreline and the contour plane, and h is the distance from the muzzle to the contour plane along a line normal to the plane.

We define $\vec{p}$ as the vector which is directed from the boreline to A and is normal to the shockwave surface. We then have

$$\vec{p} = \vec{r} - \vec{\xi} \tag{A-1}$$

where $\vec{\xi}$ is a vector along the boreline as shown in Figure A-1. The angle between $\vec{r}$ and $\vec{p}$ is designated as η . The components of $\vec{p}$ are found to be

$$p_x = x_A \tag{A-2}$$

$$p_y = y_A - \xi\cos\phi \tag{A-3}$$

$$p_z = z_A - \xi\sin\phi \quad . \tag{A-4}$$

The magnitude of $\vec{\xi}$, ξ, is found from the law of sines:

$$\frac{\sin(180^0 - \theta - \eta)}{r} = \frac{\sin \eta}{\xi} \quad . \tag{A-5}$$

Upon rearrangement, Equation (A-5) becomes

$$\xi = \frac{r \sin \eta}{\sin(\theta + \eta)} = \frac{r \sec \theta}{\frac{\tan \theta}{\tan \eta} + 1} \quad . \tag{A-6}$$

The cosine of the angle of incidence is given by

$$\cos \alpha_1 = \frac{\bar{p} \cdot \bar{n}}{p} = \frac{- z_A + \xi \sin \phi}{(p_x^2 + p_y^2 + p_z^2)^{1/2}} \tag{A-7}$$

where $\vec{n}$ is the unit normal to the contour plane. To use Equation (A-7) we must first obtain ξ . If η is known, Equation (A-6) may be employed to find ξ . The discussion below describes how η may be found.

Referring again to Figure A-1 we see that

$$\tan \eta = - \frac{1}{r} \left(\frac{dr}{d\theta}\right)_{t_a} \tag{A-8}$$

where $(dr/d\theta)_{t_a}$ denotes the derivative of r with respect to θ along a contour of time of arrival, t_a. To obtain $(dr/d\theta)_{t_a}$ the following identity is used:

$$\left(\frac{dr}{d\theta}\right)_{t_a} = - \left(\frac{\partial t_a}{\partial \theta}\right)_r \Big/ \left(\frac{\partial t_a}{\partial r}\right)_\theta \quad . \tag{A-9}$$

Recall that

$$t_a = \frac{\ell'}{a_\infty} [X\ f(Z) + C_1 + C_2 \cos\theta] \tag{A-10}$$

where

$$X = r/\ell' ,\ Z = X^{-1.1}$$

and

$$\ell' = \ell' (\theta) \quad .$$

Hence, the chain rule may be applied to yield

$$\left(\frac{\partial t_a}{\partial \theta}\right)_r = \frac{dt_a}{dX}\left(\frac{\partial X}{\partial \theta}\right)_r + \left(\frac{\partial t_a}{\partial \theta}\right)_X \qquad \text{(A-11)}$$

$$\left(\frac{\partial t_a}{\partial r}\right)_\theta = \frac{dt_a}{dX}\left(\frac{\partial X}{\partial r}\right)_\theta + \left(\frac{\partial t_a}{\partial r}\right)_X \quad . \qquad \text{(A-12)}$$

Furthermore, by differentiation, we find

$$\left(\frac{\partial X}{\partial \theta}\right)_r = -\frac{r}{(\ell')^2}\frac{d\ell'}{d\theta} \quad , \qquad \text{(A-13)}$$

$$\left(\frac{\partial X}{\partial r}\right)_\theta = \frac{1}{\ell'} \quad , \qquad \text{(A-14)}$$

$$\left(\frac{\partial t_a}{\partial \theta}\right)_X = \frac{t_a}{\ell'}\frac{d\ell'}{d\theta} - \frac{\ell'}{a_\infty} C_2 \sin\theta \; , \qquad \text{(A-15)}$$

and

$$\left(\frac{\partial t_a}{\partial r}\right)_X = 0 \quad . \qquad \text{(A-16)}$$

The following result is obtained from Reference 8:

$$\frac{dt_a}{dX} = \frac{\ell'}{a_\infty}\left(\frac{X^{1.1}}{1 + X^{1.1}}\right) \quad . \qquad \text{(A-17)}$$

Equations (A-11) and (A-12) may now be rewritten as

$$\left(\frac{\partial t_a}{\partial \theta}\right)_r = \frac{1}{a_\infty}\frac{d\ell'}{d\theta}\left[\frac{a_\infty t_a}{\ell'} - X\left(\frac{X^{1.1}}{1 + X^{1.1}}\right)\right] - \frac{\ell'}{a_\infty} C_2 \sin\theta \qquad \text{(A-18)}$$

and

$$\left(\frac{\partial t_a}{\partial r}\right)_\theta = \frac{1}{a_\infty}\left(\frac{X^{1.1}}{1 + X^{1.1}}\right) \quad . \qquad \text{(A-19)}$$

Substitution of Equations (A-18) and (A-19) into (A-9) yields

$$\left(\frac{dr}{d\theta}\right)_{t_a} = \frac{d\ell'}{d\theta}\left[X - \bar{t}_a\left(1 + X^{-1.1}\right)\right] + C_2\ell'\left(1 + X^{-1.1}\right)\sin\theta \qquad \text{(A-20)}$$

where

$$\bar{t}_a = \frac{a_\infty t_a}{\ell'} \ .$$

Finally we obtain

$$\tan\eta = \frac{1}{r}\frac{d\ell'}{d\theta}\left[\bar{t}_a\left(1 + \frac{1}{X^{1.1}}\right) - X\right] - \frac{C_2}{X}\left(1 + \frac{1}{X^{1.1}}\right)\sin\theta . \qquad \text{(A-21)}$$

In summary, to determine α_1, we first compute η from Equation (A-21). Equation (A-6) is then used to obtain ξ, which is substituted into Equations (A-2) through (A-4) to determine the components of $\vec{p}$. Finally, Equation (A-7) is used to calculate α_1. This approach is less time-consuming than direct computation of the gradient of t_a because we make use of dt_a/dX which is available to us from previous analysis.

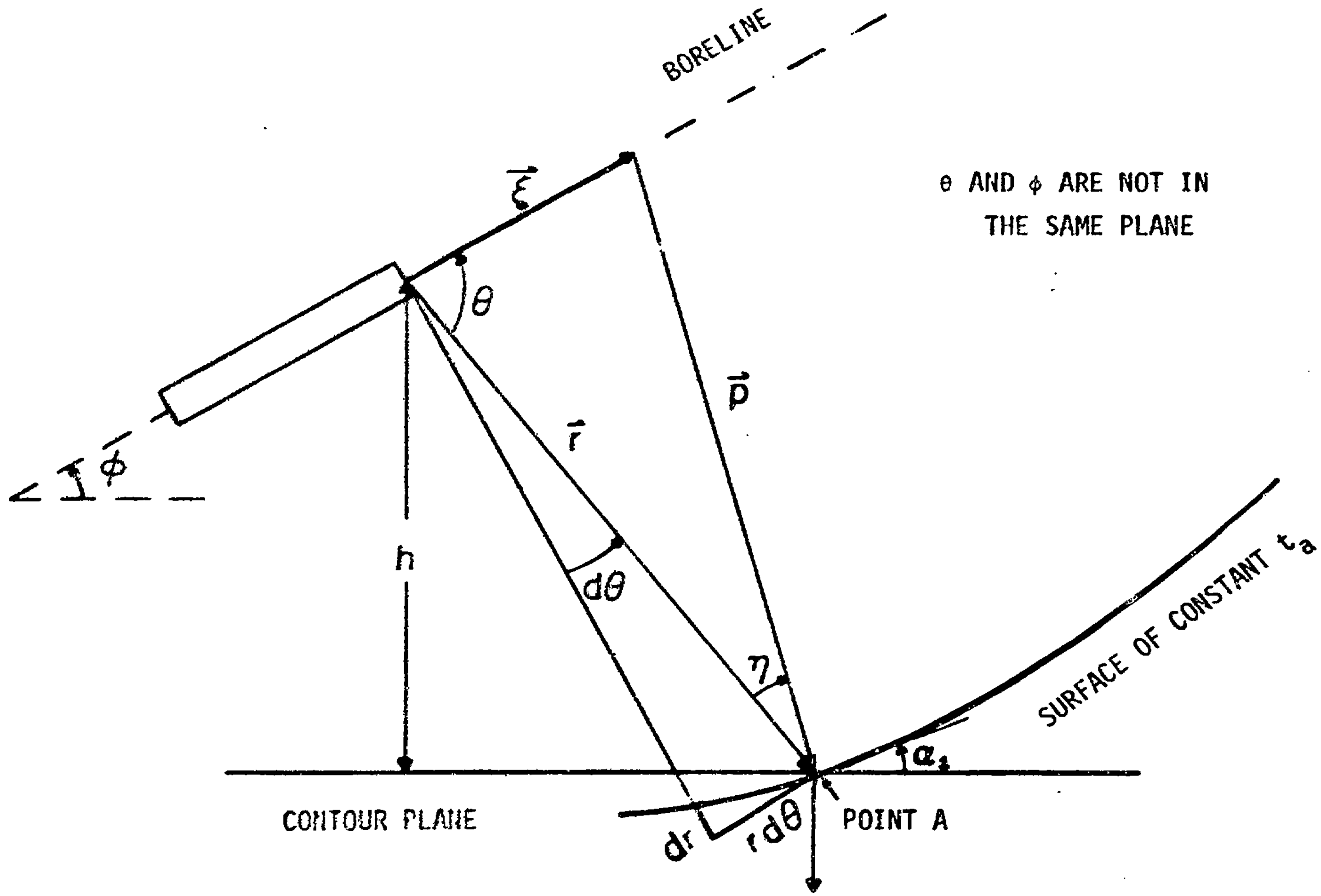


Figure A-1. Geometry of the Shock Wave Angle of Incidence Calculation

APPENDIX B - LISTING OF BLAST

```
C                      ***********************
C                      *       BLAST         *
C                      ***********************
C
C-------------------------------------------------------------------------
C       BLAST GENERATES CONTOUR MAPS OF PEAK INCIDENT OVERPRESSURE,
C PEAK REFLECTED OVERPRESSURE, BLAST WAVE TIME OF ARRIVAL, AND POSITIVE
C PHASE DURATION OF MUZZLE BLAST FROM CANNON.  CONTOURS OF THESE QUANTI-
C TIES ARE PLOTTED IN A POLAR COORDINATE SYSTEM WHICH IS LOCATED IN AN
C ARBITRARY PLANE WITH RESPECT TO THE GUN TUBE.  THE FREE FIELD BLAST
C QUANTITIES ARE COMPUTED USING SCALING RELATIONS DEVELOPED BY FANSLER
C AND SCHMIDT (ABRL-TR-02504).  THE REFLECTED OVERPRESSURE IS COMPUTED
C USING THE ANALYSIS OF HEAPS, FANSLER, AND SCHMIDT.
C THE PROGRAM IS WRITTEN IN FORTRAN 77 WITH A FEW VAX-11 FORTRAN
C STATEMENTS.  THE PLOTTING IS DONE USING THE COMMERCIAL SOFTWARE
C PACKAGE, DISSPLA.  THE PROGRAM IS DESIGNED TO BE RUN INTERACTIVELY
C ON SEVERAL TYPES OF TERMINALS WITH GRAPHICS CAPABILITY.
C
C THIS PROGRAM WAS DEVELOPED BY THE FLUID PHYSICS BRANCH OF THE LAUNCH
C AND FLIGHT DIVISION OF THE BALLISTIC RESEARCH LABORATORY.
C
C CURRENT VERSION CREATED OCTOBER 84
C-------------------------------------------------------------------------

      PROGRAM BLAST
      PARAMETER(NL=40,Q=50,P=100)

C------------------------------------------------------------------
C P AND Q ARE THE HORIZONTAL AND VERTICAL GRID DIMENSIONS,
C RESPECTIVELY.  NL IS THE MAXIMUM NUMBER OF CONTOURS THAT
C MAY BE PLOTTED.
C------------------------------------------------------------------

      LOGICAL INTBAL,GRIDUP

      REAL PSIANG(11),ANGLIN(11),PSISLP(11),SLOPES(11)
      REAL TOL,PI,RTOD,DTOR
      REAL UPBND,LOWBND,YVAR,ATM,GAMMA,AMIN,XYFAC,D,PROPM
      REAL TRAVEL,BORV,CHAMV,TOTLV,PROJ,PROBB,SPECF,TEMPK,ASUBM
      REAL PRATIO,AMBSS,VSUBP,GAMMP,MU,ELEANG,PHI,COSPHI,SINPHI
      REAL RINC,USUBY,USYBZ,X,Y,Z,ZHEIGH,MAXR
      REAL R,COSTHE,SINTHE,GM1,EXPO,RFACTR,XMASST,DSQ,DONHAF,AREAB
      REAL DIVFAC,CHIHET,ENERG,LSCALE,LPRIME,CAPX,CAPZ
      REAL PBAR(P,Q),TSUBA(P,Q),TAU(P,Q),PREFL(P,Q)
      REAL TNUM,TANETA,XCE,PSUBX,PSUBY,PSUBZ
      REAL ALPHA1,PRESIN,XI,PPMAX,CONINC,H(40),PMAX,PMIN,PRMIN
      REAL PRMAX,TSAMIN,TSAMAX,TAUMIN,TAUMAX,XSTEP,RATIO(2)
      REAL TARGX,TARGY,TARGZ,PBAR1,PBAR2,TSUBA1,TSUBA2
      REAL TAU1,TAU2,TINC,T(500),PRESS(500),TMAX,TMIN,TSUBAM

      INTEGER TRMNUM,BAUD,MAJOPT,NOGRID,MENU,UNIT
      INTEGER POPT,IOPT,KLOP,KLAP,I,J,ISIGN,ICOUNT,NLEVLS
```

```
      INTEGER FINAL,TEST,LIMIT,FOUND(40),ICNT,NPNTS

      CHARACTER ESC/27/
      CHARACTER*1 ANS,ANS1,DUMMY,WUNITS
      CHARACTER*3 ERASE
      CHARACTER*15 FNAME,WFILE
      CHARACTER*32 GUN
      CHARACTER*38 LABEL1
      CHARACTER*25 LABEL2
      CHARACTER*26 LABEL3
      CHARACTER*79 LABEL6
      CHARACTER*75 LABEL7
      CHARACTER*39 LABEL8
      CHARACTER*49 NAME
      CHARACTER*44 NAME1
      CHARACTER*45 NAME2
      CHARACTER*44 NAME3
      CHARACTER*42 NAME4

      COMMON/ONE/PBAR1,TSUBA1,TAU1
      COMMON/TWO/PBAR2,TSUBA2,TAU2
      COMMON/GRID/NLEVLS,H,K,OPEN,XM,YM,ISIGN
      COMMON/INC/RINC,MAXR
      COMMON/TRIG/RTOD,DTOR,PI
      COMMON/REFL/GAMMA,TOL,UPBND,LOWBND,XI
      COMMON/LAGRAN/PSIANG,PSISLP,ANGLIN,SLOPES
      COMMON/PRESS/PBAR,POPT,KLAP
      COMMON/ATMOS/ATM
      COMMON WORK(12000),XPLOT(250),YPLOT(250),LSAV

C-----------------------------------------------------------------------
C THE ELEMENTS OF ARRAY 'ANGLIN' ARE THE ANGLES (RADIANS) AS A FUNC-
C TION OF INCIDENT OVERPRESSURE AT WHICH THE LINEAR FIT PART OF THE
C REFLECTED OVERPRESSURE CURVE STARTS.  THE ELEMENTS OF 'PSIANG'
C ARE THE INCIDENT OVERPRESSURES CORRESPONDING TO THESE ANGLES.
C THE ELEMENTS OF ARRAY 'SLOPES' ARE THE SLOPES OF THE LINEAR APPROXI-
C MATION.  THE ELEMENTS OF 'PSISLP' ARE THE INCIDENT OVERPRESSURES
C CORRESPONDING TO THESE SLOPES.  THESE FOUR ARRAYS ARE USED IN THE
C LAGRANGIAN INTERPOLATION SUBROUTINE.
C-----------------------------------------------------------------------

      DATA PSIANG/2.,5.,10.,15.,20.,25.,30.,40.,50.,60.,70./
      DATA ANGLIN/1.29,1.15,1.06,1.01,.98,.96,.93,.91,.89,.87,.86/
      DATA PSISLP/2.,5.,10.,15.,20.,25.,30.,40.,50.,60.,70./
      DATA SLOPES/-4.1,-3.,-2.4,-2.1,-2.,-2.,-1.9,-1.8,-1.7,-1.6,
     & -1.5/

      TOL=.000001
      PI=3.141593
      RTOD=180./PI
      DTOR=PI/180.
      UPBND=PI/2
```

```
      LOWBND=.01
      YVAR=1.013*10.**5
      ATM=14.70
      GAMMA = 1.40
      AMIN=0.
      C1 = 9.24
      C2 = .94
      GRIDUP = .FALSE.

C-----------------------------------------------------------------
C  THE CURRENT VERSION HAS FIVE TERMINAL TYPE DISTINCTIONS.
C-----------------------------------------------------------------

     WRITE(6,*)
     WRITE(6,*) ' Specify the type of terminal you are using.'
     WRITE(6,*)
     WRITE(6,*) ' 0   Retrographics.'
     WRITE(6,*) ' 1   Tektronix 4010 or 4051'
     WRITE(6,*) ' 2   Tektronix 4014'
     WRITE(6,*) ' 3   Hewlett Packard 2623'
     WRITE(6,*) ' 4   ID - 100'
     WRITE(6,*)
     WRITE(6,*) ' Input corresponding integer.'

     READ*,TRMNUM
 90  IF (TRMNUM .LT. 0 .OR. TRMNUM .GT. 4) THEN
        WRITE(6,*) 'Invalid choice. Try again.'
        READ*,TRMNUM
        GOTO 90
     END IF

     IF (TRMNUM .EQ. 0 .OR. TRMNUM .EQ. 4) THEN
        BAUD = 480
     ELSE
        BAUD = 960
     END IF

     NOGRID=1
     XYFAC=1.

 105 CALL SCLEAR(TRMNUM)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,*) '                    MAIN MENU '
     WRITE(6,*)
     WRITE(6,*) '  1.  Input or read information for initial plots.'
     WRITE(6,*) '  2.  Change the input information.'
```

```
      WRITE(6,*) '  3.  Generate the plotting grid or read the grid'
      WRITE(6,*) '          from a file.'
      WRITE(6,*) '  4.  Plot contours.'
      WRITE(6,*) '  5.  Write input information to file for future use.'
      WRITE(6,*) '  6.  Save the current grid in a file for future use.'
      WRITE(6,*) '  7.  Terminate the program.'
      WRITE(6,*)
      WRITE(6,*) 'Input the corresponding integer.'
      READ*, MENU

  110 IF (MENU .LT. 1 .OR. MENU .GT. 7) THEN
         WRITE(6,*) 'Invalid response. try again.'
         READ*, MENU
         GOTO 110
      END IF

      IF (MENU .EQ. 1) GOTO 1000
      IF (MENU .EQ. 2) GOTC 2000
      IF (MENU .EQ. 3) GOTO 3000
      IF (MENU .EQ. 4) GOTO 4000
      IF (MENU .EQ. 5) GOTO 5000
      IF (MENU .EQ. 6) GOTO 3500
      IF (MENU .EQ. 7) GOTO 6000

 1000 CALL SCLEAR(TRMNUM)

      WRITE(6,*)'You have chosen to input information for initial plot'
      WRITE(6,*)
      WRITE(6,*) 'Do you wish to read the information from'
      WRITE(6,*) 'a data file? (Y or N)'
      READ(*,'(A1)') ANS
      IF (ANS .EQ. 'Y') THEN
         WRITE(6,*) 'Enter the name of the data file (15 characters)'
         READ(*,'(A15)') FNAME
         UNIT = 1
         OPEN(UNIT = 1,FILE = FNAME,READONLY,STATUS = 'OLD')
      ELSE
         UNIT = 5
      END IF

      CALL SCLEAR(TRMNUM)

      WRITE(6,*) 'Enter gun nomenclature (32 characters)'
      READ(UNIT,'(A32)') GUN
      WRITE(6,*)
      WRITE(6,*) 'Enter bore diameter of gun (meters).'
      READ(UNIT,*) D
      WRITE(6,*)
      WRITE(6,*) 'Do you wish to use interior ballistic'
      WRITE(6,*) 'formulation? (Y or N)'
      READ(UNIT,'(A1)') ANS1

```

```
      IF (ANS1 .EQ. 'Y') THEN
         INTBAL = .TRUE.
         WRITE(6,*) 'Enter propellant mass (kg).'
         READ(UNIT,*) PROPM
         WRITE(6,*)
         WRITE(6,*) 'Enter the distance the projectile '
         WRITE(6,*) 'travels inbore (meters).'
         READ(UNIT,*) TRAVEL
         BORV = .25*PI*D*D*TRAVEL
         WRITE(6,*)
         WRITE(6,*) 'Enter volume of chamber (cubic meters).'
         READ(UNIT,*) CHAMV
         TOTLV = BORV + CHAMV
         WRITE(6,*)
         WRITE(6,*) 'Enter projectile mass (kg).'
         READ(UNIT,*) PROJ
         WRITE(6,*)
         WRITE(6,*) 'Enter fraction of propellant burnt.'
         READ(UNIT,*) PROPB
         WRITE(6,*)

C----------------------------------------------------------------
C THE SPECIFIC FORCE IS EQUAL TO THE GAS COEFFICIENT TIMES
C THE ADIABATIC FLAME TEMPERATURE OF THE PROPELLANT.
C----------------------------------------------------------------

         CALL SCLEAR(TRMNUM)
         WRITE(6,*)'Enter specific force. Use MKS units (m*m/s*s) (A),'
         WRITE(6,*)'or English units (ft*lb/lbm) (B). Enter A or B.'
         WRITE(6,*)
         READ(UNIT,'(A1)') WUNITS
         WRITE(6,*)
         WRITE(6,*) 'Now enter specific force.'
         WRITE(6,*)
         READ(UNIT,*) SPECF
         IF (WUNITS .EQ. 'B') SPECF = SPECF*2.992
         WRITE(6,*)
         WRITE(6,*) 'Enter flame temperature (deg. Kelvin).'
         WRITE(6,*) 'Default = 3000 K.'
         READ(UNIT,'(F12.4)') TEMPK
         IF (TEMPK .EQ. 0.0) TEMPK = 3000.0
         CALL SCLEAR(TRMNUM)
      ELSE
         INTBAL = .FALSE.
         WRITE(6,*)
         WRITE(6,*) 'Enter pre-uncorked sound speed at muzzle (m/s).'
         READ(UNIT,*) ASUBM
         WRITE(6,*)
         WRITE(6,*) 'Enter ratio of pre-uncorked muzzle pressure'
         WRITE(6,*) 'to ambient pressure.'
         READ(UNIT,*) PRATIO
      END IF
```

```
      WRITE(6,*)
      WRITE(6,*) 'Enter ambient sound speed (m/s). Default = 340.0 m/s'
      READ(UNIT,'(F12.4)') AMBSS
      IF (AMBSS .EQ. 0.0) AMBSS = 340.0

      WRITE(6,*)
      WRITE(6,*) 'Enter exit velocity of projectile.'
      READ(UNIT,*) VSUBP

      WRITE(6,*)
      WRITE(6,*) 'Enter propellent specific heat ratio, gamma.'
      WRITE(6,*) 'Default = 1.24.'
      READ(UNIT,'(F7.4)') GAMMP
      IF (GAMMP .EQ. 0.0) GAMMP = 1.24

      WRITE(6,*)
      WRITE(6,*) 'Enter momentum index, mu. Default = 0.78'
      READ(UNIT,'(F7.4)') MU
      IF (MU .EQ. 0.0) MU = 0.78

      CALL SCLEAR(TRMNUM)

      WRITE(6,*) 'Enter elevation angle of gun (degrees).'
      READ(UNIT,*) ELEANG
      PHI = DTOR*ELEANG
      COSPHI = COS(PHI)
      SINPHI = SIN(PHI)

      WRITE(6,*) 'Enter the height of the muzzle (meters).'
      READ(UNIT,*) ZHEIGH

      CLOSE (UNIT=1)

      GOTO 105

2000  CALL SCLEAR(TRMNUM)

      WRITE(6,*) 'You have chosen to change information.'
      WRITE(6,*)
      WRITE(6,*) '       Change of information menu'
      WRITE(6,*)
      WRITE(6,*) ' 1.  Change weapon and flow parameters.'
      WRITE(6,*) ' 2.  Change the orientation of the gun.'
      WRITE(6,*) ' 3.  Return to the main menu.'
      WRITE(6,*)
      WRITE(6,*) 'Enter the corresponding integer.'
      READ*,MENU
2010  IF ( MENU .LT. 1 .OR. MENU .GT. 3 ) THEN
         WRITE(6,*) 'Invalid response. Try again.'
         READ*,MENU
         GOTO 2010
```

```
      END IF

      IF (MENU .EQ. 1) GOTO 2100
      IF (MENU .EQ. 2) GOTO 2200
      IF (MENU .EQ. 3) GOTO 105

2100  CALL SCLEAR(TRMNUM)

      WRITE(6,*) 'You may change any of the following:'
      WRITE(6,*)
      WRITE(6,*) ' 1. Gun nomenclature    10. Pre-uncorked sound speed '
      WRITE(6,*) ' 2. Bore diameter            at the muzzle '
      WRITE(6,*) ' 3. Propellant mass *   11. Ratio of pre-uncorked muz-'
      WRITE(6,*) ' 4. Inbore travel            zle pressure to ambient '
      WRITE(6,*) '      distance      *        pressure '
      WRITE(6,*) ' 5. Chamber volume  *   12. Ambient sound speed'
      WRITE(6,*) ' 6. Projectile mass *   13. Projectile exit velocity'
      WRITE(6,*) ' 7. Fraction of pro-    14. Specific heat ratio'
      WRITE(6,*) '      pellant burnt *   15. Momentum index'
      WRITE(6,*) ' 8. Specific force  *'
      WRITE(6,*) ' 9. Flame temperature *'
      WRITE(6,*)
      WRITE(6,*) ' or you may             16. Return to preceeding menu'
      WRITE(6,*)
      WRITE(6,*) '* indicates an interior ballistics parameter.'
      WRITE(6,*)
      WRITE(6,*) 'Enter corresponding integer.'
      READ*,MENU

2110  IF (MENU .LT. 1 .OR. MENU .GT. 16) THEN
         WRITE(6,*) 'Invalid response. Try again.'
         READ*,MENU
         GOTO 2110
      END IF

      CALL SCLEAR(TRMNUM)

      IF (MENU .EQ. 1) THEN
         WRITE(6,*) 'Current value is ',GUN
         WRITE(6,*) 'Enter new value'
         READ(*,'(A32)') GUN
      END IF

      IF (MENU .EQ. 2) THEN
         WRITE(6,*) 'Current value is ',D
         WRITE(6,*) 'Enter new value'
         READ*,D
      END IF

      IF (MENU .EQ. 3) THEN
         WRITE(6,*) 'Current value is ',PROPM
         WRITE(6,*) 'Enter new value'
```

```
            READ*,PROPM
            ANS1 = 'Y'
         END IF

         IF (MENU .EQ. 4) THEN
            WRITE(6,*) 'Current value is ',TRAVEL
            WRITE(6,*) 'Enter new value'
            READ*,TRAVEL
            BORV = 0.25*PI*D*D*TRAVEL
            ANS1 = 'Y'
         END IF

         IF (MENU .EQ. 5) THEN
            WRITE(6,*) 'Current value is ',CHAMV
            WRITE(6,*) 'Enter new value'
            READ*,CHAMV
            ANS1 = 'Y'
         END IF

         IF (MENU .EQ. 6) THEN
            WRITE(6,*) 'Current value is ',PROJ
            WRITE(6,*) 'Enter new value'
            READ*,PROJ
            ANS1 = 'Y'
         END IF

         IF (MENU .EQ. 7) THEN
            WRITE(6,*) 'Current value is ',PROPB
            WRITE(6,*) 'Enter new value'
            READ*,PROPB
            ANS1 = 'Y'
         END IF

         IF (MENU .EQ. 8) THEN
            WRITE(6,*) 'Current value is ',SPECF
            WRITE(6,*)'Use MKS units (m*m/s*s) (A), or English'
            WRITE(6,*)'units (ft*lb/lbm) (B). Enter A or B.'
            WRITE(6,*)
            READ(*,'(A1)') WUNITS
            WRITE(6,*)
            WRITE(6,*) 'Now enter new value.'
            READ*, SPECF
            IF (WUNITS .EQ. 'B') SPECF = SPECF*2.992
            ANS1 = 'Y'
         END IF

         IF (MENU .EQ. 9) THEN
            WRITE(6,*) 'Current value is ',TEMPK
            WRITE(6,*) 'Enter new value'
            READ*,TEMPK
            ANS1 = 'Y'
         END IF
```

```
      IF (MENU .EQ. 10) THEN
         WRITE(6,*) 'Current value is ',ASUBM
         WRITE(6,*) 'Enter new value'
         READ*,ASUBM
      END IF

      IF (MENU .EQ. 11) THEN
         WRITE(6,*) 'Current value is ',PRATIO
         WRITE(6,*) 'Enter new value'
         READ*,PRATIO
      END IF

      IF (MENU .EQ. 12) THEN
         WRITE(6,*) 'Current value is ',AMBSS
         WRITE(6,*) 'Enter new value'
         READ*,AMBSS
      END IF

      IF (MENU .EQ. 13) THEN
         WRITE(6,*) 'Current value is ',VSUBP
         WRITE(6,*) 'Enter new value'
         READ*,VSUBP
      END IF

      IF (MENU .EQ. 14) THEN
         WRITE(6,*) 'Current value is ',GAMMP
         WRITE(6,*) 'Enter new value'
         READ*,GAMMP
      END IF

      IF (MENU .EQ. 15) THEN
         WRITE(6,*) 'Current value is ',MU
         WRITE(6,*) 'Enter new value'
         READ*,MU
      END IF

      IF (MENU .EQ. 16) GOTO 2000
      GOTO 2100

 2200 CALL SCLEAR(TRMNUM)
      WRITE(6,*) 'Current elevation angle of gun is ',ELEANG,
     &  ' degrees.'
      WRITE(6,*)
      WRITE(6,*) 'Enter the new elevation angle.'
      READ*, ELEANG
      PHI = DTOR*ELEANG
      COSPHI = COS(PHI)
      SINPHI = SIN(PHI)

      WRITE(6,*)
      WRITE(6,*) 'The current height of the muzzle is ',ZHEIGH
```

```
      WRITE(6,*)
      WRITE(6,*) 'Enter the new height of the muzzle.'
      READ*, ZHEIGH

      GOTO 2000

C-----------------------------------------------------------------------
C  IN SECTION 3000 THE USER CHOOSES WHICH QUANTITY HE WANTS PLOTTED.
C  THE VALUES OF THIS QUANTITY ARE THEN DETERMINED AT EACH GRID POINT
C  IN THE CONTOUR PLANE.
C-----------------------------------------------------------------------

3000  CALL SCLEAR(TRMNUM)

C---------------------------------------------------------------------
C THE USER MAY GENERATE A GRID TO PLOT ALL QUANTITIES, OR HE MAY
C CHOOSE TO EXCLUDE THE REFLECTED OVERPRESSURE.  THIS CHOICE IS
C GIVEN BECAUSE THE REFLECTED OVERPRESSURE GRID IS MUCH MORE TIME-
C CONSUMING TO GENERATE THAN THE OTHER QUANTITIES.
C THE USER MAY ALSO READ A GRID THAT WAS PREVIOUSLY GENERATED.
C---------------------------------------------------------------------

       WRITE(6,*)
       WRITE(6,*)'You may'
       WRITE(6,*)
       WRITE(6,*)' 1.  Generate a grid of values for '
       WRITE(6,*)'         peak incident overpressure,'
       WRITE(6,*)'         peak reflected overpressure,'
       WRITE(6,*)'         blast wave time of arrival,'
       WRITE(6,*)'         positive phase duration.'
       WRITE(6,*)
       WRITE(6,*)' 2.  Generate a grid of values for all of the'
       WRITE(6,*)'     quantities above excluding peak reflected'
       WRITE(6,*)'     overpressure.'
       WRITE(6,*)
       WRITE(6,*)' 3.  Read a previously generated grid from a'
       WRITE(6,*)'     data file.'
       WRITE(6,*)
       WRITE(6,*)' Enter the corresponding integer. '
       READ *,IOPT

       IF (IOPT .NE. 3) THEN

C-----------
C NEW PAGE.
C-----------

       CALL SCLEAR(TRMNUM)
       KLOP=0
       KLAP=0

C---------------------------------------------------------------------
```

```
C THE USER DEFINES THE MAXIMUM RADIUS TO BE PLOTTED.
C-----------------------------------------------------------------------

      WRITE(6,*)'Input maximum distance from origin to be plotted
     & (meters).'
      WRITE(6,*)'The origin is the intersection point of the contour
     & plane and'
      WRITE(6,*)'the line that passes through the muzzle of the gun
     & and'
      WRITE(6,*)'is perpendicular to the contour plane.'
      WRITE(6,*)
      READ *,MAXR

C-----------------------------------------------------------------------
C PHI IS THE ANGLE BETWEEN THE BORELINE AND THE PLANE IN WHICH
C THE CONTOURS ARE TO BE PLOTTED.  USUBY AND USUBZ ARE THE Y AND Z
C COMPONENTS OF THE UNIT VECTOR, U, PARALLEL TO THE BORELINE OF THE
C GUN.  THE FUNCTION TO BE PLOTTED IS EVALUATED AT EACH GRIDPOINT.
C THIS PROCESS PROCEEDS ROW BY ROW STARTING IN THE LOWER LEFT CORNER
C OF THE GRID.  THE GRID IS A RECTANGLE WHICH EXTENDS FROM 0 TO TWO
C TIMES MAXR HORIZONTALLY AND FROM 0 TO MAXR VERTICALLY.THE MAXIMUM
C RADIUS IS ROUNDED TO HUNDREDS IN ORDER TO FIT ON THE PLOT PROPERLY.
C-----------------------------------------------------------------------

      MAXR=INT(MAXR*P)/P

      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      IF (IOPT.EQ.1) THEN
         WRITE(6,*)'Working.  The calculation of functional values will
     & take several minutes.'
         WRITE(6,*)'Wait for bell.'
      END IF
      IF (IOPT .EQ. 2) THEN
         WRITE(6,*)'Working.  Wait for bell.'
      END IF

      RINC=2.*MAXR/P

      USUBY=COSPHI
      USUBZ=SINPHI

      DO 3160 J=1,Q
           DO 3165 I=1,P

C-----------------------------------------------------------------------
C X, Y, AND Z ARE THE X, Y AND Z COMPONENTS OF THE VECTOR, R,
```

```
C WHICH POINTS FROM THE ORIGIN OF THE CONTOUR PLANE TO THE POINT IN
C THE CONTOUR GRID UNDER CONSIDERATION.  X,Y, AND Z ARE THE COORDI-
C NATES OF THIS GRIDPOINT REFERRED TO AN ORIGIN AT THE MUZZLE.  THETA
C IS THE POLAR ANGLE OF THE VECTOR R.  THETA IS DETERMINED BY
C TAKING THE DOT PRODUCT OF R AND U.
C-----------------------------------------------------------------------

      X=-J*RINC
      IF (I*RINC.LT.MAXR) Y=-(MAXR-I*RINC)
      IF (I*RINC.EQ.MAXR) Y=0.
      IF (I*RINC.GT.MAXR) Y=I*RINC-MAXR
      Z=-ZHEIGH
      R=SQRT(X**2 + Y**2 + Z**2)
      COSTHE=((Y*USUBY)+(Z*USUBZ))/R
      SINTHE=SQRT(1. - COSTHE**2)
      GM1=GAMMP-1.
      EXPO=(3.*GAMMP-1.)/GM1

C-------------------------------------------------------------
C THE SCALING LENGTHS, LSCALE AND LPRIME, ARE DETERMINED.
C FIRST, INTERIOR BALLISTIC PARAMETERS ARE CONSIDERED.
C-------------------------------------------------------------

      IF (INTBAL) THEN
         PROJM = 1.05*PROJ
         RFACTR = 1.35
         IF(D.LT.0.02) RFACTR = 1.5
         IF(D.GT.0.045) RFACTR = 1.3
         XMASST = PROJM + PROPM/3.
         DSQ = D**2
         DONHAF = SQRT(D)
         AREAB = PI*(D**2)/4.
         DIVFAC = 1.7 + 671.*DONHAF*((DSQ/PROPM)**.86)
         CHIHET=10500.*TOTLV*D*DONHAF*(TEMPK-300.)*RFACTR/AREAB
     &    /(VSUBP**2)/XMASST/DIVFAC
         ENERG=((PROPB*PROPM*SPECF)/GM1)-.5*(PROJM+
     &   PROPM/3.)*(1.0+CHIHET)*VSUBP**2
         ASUBM=((GAMMP*GM1*ENERG)/(PROPM+(PROPM**2/(3.*PROJM
     &   ))))**.5
         PRATIO=(GM1*ENERG)/(TOTLV*(1+PROPM/(3.*PROJM)))/YVAR
      END IF

C-----------------------------------------------------------------------
C NEXT MUZZLE EXIT CONDITIONS FOR SUBSONIC EXIT FLOW.
C-----------------------------------------------------------------------

         IF (VSUBP.LE.ASUBM) THEN
            LSCALE=D*SQRT((.00862*PRATIO*ASUBM/(GM1*AMBSS))
     &      *(1.+GAMMP*GM1/2.)*((2.+GM1*VSUBP/ASUBM)/
     &      (GAMMP+1.))**EXPO)
         ELSE

```

```
C-----------------------------------------------------------------------
C FINALLY, MUZZLE EXIT CONDITIONS FOR SUPERSONIC EXIT FLOW.
C-----------------------------------------------------------------------

                  LSCALE=D*.0928*SQRT((PRATIO*VSUBP/(GM1*AMBSS))*
     &            (1.+GAMMP*GM1*.5*(VSUBP/ASUBM)**2))
                END IF

C-----------------------------------------------------------------------
C LPRIME IS CALCULATED.
C-----------------------------------------------------------------------

        LPRIME=LSCALE*((MU*COSTHE)+SQRT(1.-(MU**2)*(SINTHE**2)))
        CAPX=R/LPRIME
        CAPZ=CAPX**-1.1
        IF (CAPX.LT.1.) KLOP=1

C-------------------------------------------------------------------
C PBAR IS THE PEAK INCIDENT OVERPRESSURE, TSUBA IS THE BLAST WAVE
C TIME OF ARRIVAL, AND TAU IS THE POSITIVE PHASE DURATION.  THESE
C QUANTITIES ARE CALCULATED USING THE SCALING RELATIONS DERIVED BY
C FANSLER AND SCHMIDT.NEW PBAR AND TSUBA EQUATIONS ADDED MARCH,83.
C-------------------------------------------------------------------

        PBAR(I,J)=2.4*CAPZ

        TSUBA(I,J)=R/AMBSS*(1.+10.*CAPZ-.8333*CAPZ**2+0.4348*
     &  CAPZ**3-.2941*CAPZ**4+.2273*CAPZ**5-.1667*CAPZ**6)-
     &  (LPRIME/AMBSS)*(C1+C2*COSTHE)

        TAU(I,J)=(LPRIME/AMBSS)*(1.0 +  .13*CAPX)

C-----------------------------------------------------------------
C IF IOPT=1 WE GO THROUGH THE REFLECTED OVERPRESSURE CALCULATION.
C THE FIRST STEP IS TO DETERMINE THE SHOCK WAVE ANGLE OF INCIDENCE
C AT EACH GRID POINT.
C-----------------------------------------------------------------

        IF (IOPT .EQ. 1) THEN

C-----------------------------------------------------------------
C ALPHA1 IS THE SHOCK WAVE ANGLE OF INCIDENCE AT THE GRIDPOINT.
C A DETAILED EXPLANATION OF THE ALPHA1 CALCULATION IS IN APPENDIX
C A OF THE REPORT.
C------------------------------------------------------------------

        TSABAR = TSUBA(I,J)*AMBSS/LPRIME

C------------------------------------------------------------------
C DLPDTH = DERIVATIVE OF LPRIME WITH RESPECT TO THETA
C DCXDTB = DERIVATIVE OF CAPX WITH RESPECT TO TSUBA BAR
C------------------------------------------------------------------
```

```
            DLPDTH = -LPRIME*MU*SINTHE/SQRT(1. - MU*MU*SINTHE**2)
            DCXDLP = -R/LPRIME**2

C-------------------------------------------------------------------------
C IF THE TIME OF ARRIVAL EXPRESSION IS CHANGED, DTBDCX MUST ALSO BE
C CHANGED ACCORDINGLY.
C-------------------------------------------------------------------------

            DCXDTB = 1. + CAPZ

            TANETA = (DLPDTH/LPRIME)*( DCXDTB*TSABAR/CAPX - 1.0 )
     &              + C2*LPRIME*DCXDTB*SINTHE/CAPX

            IF (COSTHE .EQ. 0.) THEN
               XCE = R*TANETA
            ELSE
               TANTHE = SINTHE/COSTHE
               XCE = (R/COSTHE)/(TANTHE/TANETA + 1.0)
            END IF

            PSUBX = X
            PSUBY = Y - XCE*COSPHI
            PSUBZ = Z - XCE*SINPHI
            ALPHA1 = ACOS(-PSUBZ/SQRT(PSUBX**2+PSUBY**2+PSUBZ**2))
            IF (ALPHA1 .GT. PI/2.) ALPHA1 = PI - ALPHA1
            PRESIN=PBAR(I,J)
            XI=PRESIN+1.

C-----------------------------------------------------------------
C SUBROUTINE REFCAL IS CALLED TO DETERMINE THE REFLECTED
C OVERPRESSURE, PREFL, AT THE GRIDPOINT.
C IF THE ANGLE OF INCIDENCE IS GREATER THAN 1. DEGREE WE USE
C OBLIQUE SHOCK THEORY.  IF ANGLE OF INCIDENCE IS LESS THAN
C 1. DEGREE WE USE NORMAL SHOCK RELATIONS.
C-----------------------------------------------------------------

            IF (ALPHA1 .GT. 0.0174) THEN
               CALL REFCAL(PRESIN,PREFL(I,J),ALPHA1)
            ELSE
               P3 = ( XI**2*(3*GAMMA-1.) - (GAMMA-1.)*XI )/
     &                ( XI*(GAMMA-1.) + (GAMMA+1.) )
               PREFL(I,J) = P3 - 1.
            END IF

            END IF
3165        CONTINUE
3160      CONTINUE
          CALL BELL
          CALL BELL
          GRIDUP = .TRUE.
          GOTO 105
```

```
      ELSE
C-------------------------------------------------------------------
C  IF THE USER HAS CHOSEN TO READ A GRID THAT HAS BEEN SAVED
C  THE GRID IS NOW READ FROM THE DATA FILE.
C-------------------------------------------------------------------
3200  CALL SCLEAR(TRMNUM)
      WRITE(6,*) 'What is the name of the data file?'
      READ(*,'(A20)') FNAME
      OPEN(UNIT=1,FILE=FNAME,STATUS='OLD')
      WRITE(6,*) 'Reading grid from file ',FNAME
      READ(1,*) IOPT
      READ(1,*) MAXR
      IF (IOPT .EQ. 1) THEN
         DO 3210 I=1,P
            DO 3215 J=1,Q
               READ(1,*) PBAR(I,J),PREFL(I,J),TSUBA(I,J),TAU(I,J)
3215        CONTINUE
3210     CONTINUE
      END IF
      IF (IOPT .EQ. 2) THEN
         DO 3220 I=1,P
            DO 3225 J=1,Q
               READ(1,*) PBAR(I,J),TSUBA(I,J),TAU(I,J)
3225        CONTINUE
3220     CONTINUE
      END IF
      CLOSE(UNIT=1)
      RINC=2.*MAXR/P
      CALL BELL
      CALL BELL
      CALL BELL
      GRIDUP = .TRUE.
      GOTO 105
      END IF
C-------------------------------------------------------------------
C  THE USER CAN WRITE THE CURRENT GRID TO A DATA FILE FOR FUTURE
C  USE TO AVOID SPENDING THE TIME GENERATING IT.
C-------------------------------------------------------------------
3500  CALL SCLEAR(TRMNUM)
      WRITE(6,*)
      WRITE(6,*) 'Enter the name you wish to give the file.'
      READ(*,'(A20)') FNAME
      OPEN(UNIT=1,FILE=FNAME,STATUS='NEW')
      WRITE(6,*)
      WRITE(6,*) 'Writing grid to file ',FNAME
```

```
      WRITE(1,*) IOPT
      WRITE(1,*) MAXR
      IF (IOPT .EQ. 1) THEN
         DO 3505 I=1,P
            DO 3510 J=1,Q
               WRITE(1,*) PBAR(I,J),PREFL(I,J),TSUBA(I,J),TAU(I,J)
3510        CONTINUE
3505     CONTINUE
      END IF
      IF (IOPT .EQ. 2) THEN
         DO 3520 I=1,P
            DO 3530 J=1,Q
               WRITE(1,*) PBAR(I,J),TSUBA(I,J),TAU(I,J)
3530        CONTINUE
3520     CONTINUE
      END IF
      CLOSE(UNIT=1)
      CALL BELL
      CALL BELL
      GOTO 105

C-----------------------------------------------------------------------
C IN SECTION 4000 THE USER PICKS WHICH QUANTITY HE WANTS TO PLOT.
C THE GRID USED MAY BE ONE GENERATED DURING THE CURRENT INTERACTIVE
C SESSION OR IT MAY BE ONE WHICH SAVED FROM A PREVIOUS SESSION.
C THIS FEATURE ALLOWS THE USER TO RUN THE TIME-CONSUMING PART
C OF THE PROGRAM ON A NON-GRAPHICS TERMINAL AND THEN MOVE TO A
C GRAPHICS TERMINAL TO QUICKLY GENERATE THE CONTOUR MAPS.
C THE CONTOUR LEVELS TO BE PLOTTED ARE SPECIFIED
C BY THE USER OR DETERMINED AUTOMATICALLY.  THE CONTOURING ALGOR-
C ITHM THEN SEARCHES THROUGH THE GRID TO DETERMINE THE PATH OF
C EACH CONTOUR LEVEL.  ONCE THE PATH HAS BEEN DETERMINED, IT IS
C PLOTTED USING THE DISSPLA SUBROUTINES.
C-----------------------------------------------------------------------

4000  CALL SCLEAR(TRMNUM)
      IF (.NOT. GRIDUP) THEN
         WRITE(6,*) 'You must generate or read a grid of function'
         WRITE(6,*) 'values before plotting. Hit return to return'
         WRITE(6,*) 'to main menu.  Then choose menu option 3.'
         READ(*,'(F3.1)') PAUS
         GOTO 105
      END IF

C------------------------------
C THE LABELS ARE DEFINED.
C------------------------------

      LABEL1='DISTANCE FROM ORIGIN OF PLANE, meters '
      LABEL2='ANGLE OF ELEVATION (deg)='
      LABEL3='HEIGHT OF MUZZLE (meters)='
      LABEL6='YOUR MAXIMUM CONTOUR VALUE IS TOO LARGE, PLEASE
```

```
     & TRY AGAIN. IT MUST NOT EXCEED -'
       LABEL7='YOUR MAXIMUM CONTOUR VALUE IS TOO SMALL, PLEASE
     & TRY AGAIN. IT MUST EXCEED -'
       LABEL8='INPUT NEW VALUE (ENTER 0. FOR DEFAULT).'

       NAME1='CONTOURS OF PEAK INCIDENT OVERPRESSURE, mbar'
       NAME2='CONTOURS OF PEAK REFLECTED OVERPRESSURE, mbar'
       NAME3='CONTOURS OF BLAST WAVE TIME OF ARRIVAL, msec'
       NAME4='CONTOURS OF POSITIVE PHASE DURATION, msec'

       WRITE (6,*)
       WRITE (6,*) '                PLOTTING MENU'
       WRITE (6,*)
       WRITE (6,*) ' 1.  Plot contours.'
       WRITE (6,*) ' 2.  Enlarge or reduce plot size.'
       WRITE (6,*) ' 3.  Deactivate/activate polar grid on plot.'
       WRITE (6,*) ' 4.  Return to preceeding menu.'
       WRITE (6,*)
       WRITE (6,*) 'Enter the corresponding integer.'
       READ*,MENU

       IF (MENU .EQ. 1) GOTO 4100
       IF (MENU .EQ. 2) GOTO 4200
       IF (MENU .EQ. 3) GOTO 4300
       IF (MENU .EQ. 4) GOTO 105

C-----------------------------------------------------------------------
C IF THE USER WISHES TO PLOT CONTOURS ON A GRID THAT WAS PREVIOUSLY
C SAVED IN A DATA FILE HE MAY SPECIFY THE DATA FILE NAME.
C-----------------------------------------------------------------------

4100   CALL SCLEAR(TRMNUM)

C-----------------------------------------------------------------------
C THE USER CHOOSES WHICH BLAST QUANTITY HE WANTS TO PLOT.
C-----------------------------------------------------------------------

       WRITE(6,*)
       WRITE(6,*)' Choose which quantity you wish to plot:'
       WRITE(6,*)
       IF (IOPT .EQ. 1) THEN
          WRITE(6,*)'  1. Peak incident overpressure.'
          WRITE(6,*)'  2. Peak reflected overpressure.'
          WRITE(6,*)'  3. Blast wave time of arrival.'
          WRITE(6,*)'  4. Blast wave positive phase duration.'
          WRITE(6,*)
          READ*,POPT
       ELSE
          WRITE(6,*)'  1. Peak incident overpressure.'
          WRITE(6,*)'  2. Blast wave time of arrival.'
          WRITE(6,*)'  3. Blast wave positive phase duration.'
          WRITE(6,*)
```

```
         READ*,POPT
         IF (POPT .EQ. 3) POPT=4
         IF (POPT .EQ. 2) POPT=3
      END IF

C-----------------------------------------------------------------
C THE USER CHOOSES WHETHER HE WANTS TO SPECIFY THE VALUE OF EACH
C CONTOUR OR HAVE THE CONTOUR LEVELS AUTOMATICALLY SCALED STARTING
C WITH A COMPUTER GENERATED MINIMUM VALUE.
C-----------------------------------------------------------------

      CALL SCLEAR(TRMNUM)
      WRITE(6,*)'You may'
      WRITE(6,*)
      WRITE(6,*)'  1.  Enter the contour values to be plotted.'
      WRITE(6,*)'  2.  Have the contour values automatically'
      WRITE(6,*)'      calculated based on the minimum and maximum'
      WRITE(6,*)'      values of the function in the plotting domain.'
      WRITE(6,*)
      READ*, MENU
      IF (MENU.EQ.1) THEN

      CALL SCLEAR(TRMNUM)
      WRITE(6,*)'Input contour values to be plotted (mbar or msec,'
      WRITE(6,*)'max of 40 values)'
      WRITE(6,*)
      WRITE(6,*)'If no more inputs, enter 0. '
4101  FORMAT(' ','Contour ',1X,I2)
      ICOUNT=0
      DO 4135 I=1,40
         WRITE(6,*)
         WRITE(6,4101) I
         READ *, H(I)
         IF (POPT.EQ.1.AND.H(I).GE.2.4) THEN
            KLAP=1
         ELSE
            KLAP=0
         END IF
         H(I) = H(I)/1000.
         IF(H(I).EQ.0.0) THEN
            NLEVLS=ICOUNT
            GO TO 4136
         ELSE
            ICOUNT=ICOUNT+1
         END IF
4135  CONTINUE

C-----------------------------------------------------------
C IF THE USER INPUTS THE CONTOUR VALUES HIMSELF, THEY ARE
C SORTED INTO INCREASING ORDER.
C-----------------------------------------------------------

```

```
 4136 FINAL=NLEVLS
      TEST=NLEVLS-1
      DO 4140 J=1,TEST
         LIMIT=FINAL-1
         DO 4145 I=1,LIMIT
            IF (H(I).GT.H(I+1)) THEN
               TEMP=H(I)
               H(I)=H(I+1)
               H(I+1)=TEMP
            END IF
 4145    CONTINUE
         FINAL=FINAL-1
 4140   CONTINUE

      ELSE

C------------------------------------------------------------
C IF THE CONTOUR VALUES ARE TO BE COMPUTER CHOSEN, THE MAX-
C IMUM AND MINIMUM VALUES IN THE CONTOUR GRID ARE FOUND AND
C THE CONTOUR LEVELS ARE SCALED ACCORDING TO THE INCREMENT.
C------------------------------------------------------------

      WRITE(6,*)
      WRITE(6,*) 'Enter the number of contours you want plotted'
      WRITE(6,*) '(maximum of 40).'
      READ*,NLEVLS
      WRITE(6,*)

 4250 IF (POPT.EQ.1) THEN
         CALL FNDMAX(PBAR,PMAX,POPT)
         CALL FNDMIN(PBAR,PMIN,POPT)
         PMIN = TWOSIG(1.1*PMIN)
         PMAX = .9*PMAX
         CINC=(PMAX-PMIN)/(NLEVLS-1)
         CINC=ONESIG(CINC)
         DO 4255 I=1,NLEVLS
            H(I)=PMIN + (I-1)*CINC
            IF (H(I) .GT. 2.4) THEN
               KLAP = 1
            ELSE
               KLAP = 0
            END IF
  4255   CONTINUE
      END IF
      IF (POPT.EQ.2) THEN
         CALL FNDMAX(PREFL,PRMAX,POPT)
         CALL FNDMIN(PREFL,PRMIN,POPT)
         PRMIN = TWOSIG(1.1*PRMIN)
         PRMAX = 0.9*PRMAX
         CINC=(PRMAX-PRMIN)/(NLEVLS-1)
         CINC=ONESIG(CINC)
         DO 4260 I=1,NLEVLS
```

```
               H(I)=PRMIN + ((I-1)*CINC)
 4260       CONTINUE
         END IF
         IF (POPT.EQ.3) THEN
            CALL FNDMAX(TSUBA,TSAMAX,POPT)
            CALL FNDMIN(TSUBA,TSAMIN,POPT)
            TSAMIN = TSAMIN*1.1
            TSAMAX = TWOSIG(TSAMAX*.9)
            CINC=(TSAMAX-TSAMIN)/(NLEVLS-1)
            CINC=ONESIG(CINC)
            DO 4265 I=1,NLEVLS
               H(I)=TSAMAX-((I-1)*CINC)
 4265       CONTINUE
         END IF
         IF (POPT.EQ.4) THEN
            CALL FNDMAX(TAU,TAUMAX,POPT)
            CALL FNDMIN(TAU,TAUMIN,POPT)
            TAUMIN=TAUMIN*1.1
            TAUMAX=TWOSIG(TAUMAX*.9)
            CINC=(TAUMAX-TAUMIN)/(NLEVLS-1)
            CINC=ONESIG(CINC)
            DO 4267 I=1,NLEVLS
               H(I)=TAUMAX-((I-1)*CINC)
  4267      CONTINUE
         END IF

         END IF

C-----------------------------------------------------------------
C FOUND(I)=0 MEANS THAT NO CONTOURS WERE LOCATED AT THE H(I)
C LEVEL.  FOUND(I)=1 MEANS THAT CONTOURS WERE LOCATED.
C-----------------------------------------------------------------

         DO 4173 I=1,NLEVLS
            FOUND(I)=0
 4173    CONTINUE

C-----------------------------------------------------------------
C THE USER'S TERMINAL IS READIED FOR THE PLOTS .THE CONTOURS
C WILL BE DRAWN USING DISSPLA VERSION 9.0 COMMANDS. BECAUSE
C OF THE SYMMETRY OF THE PROBLEM, THETA NEED ONLY RANGE FROM
C 0 TO 180 DEGREES.
C-----------------------------------------------------------------

         IF (TRMNUM.EQ.0 .OR. TRMNUM .EQ. 4) THEN
            CALL RETRO (BAUD)
         ELSE IF (TRMNUM.EQ.1) THEN
            CALL TK4010 (BAUD)
         ELSE IF (TRMNUM.EQ.2) THEN
            CALL TK4014 (BAUD,1)
         ELSE
            CALL HP2623 (BAUD)
```

```
      END IF
      XSTEP=MAXR/5.

C-----------------------------------------------------------
C THE DISSPLA COMMANDS READY THE TERMINAL FOR PLOTTING
C BY FIRST DEFINING THE SUBPLOT AREA THEN CALIBRATING IT
C FOR THE CONTOUR LINES.
C-----------------------------------------------------------

      CALL BGNPL(0)
      CALL NOCHEK
      CALL NOBRDR
      CALL SIMPLX
      CALL PHYSOR(.1,1.0)
      CALL BLOWUP(XYFAC)
      CALL TITLE(' $',-100,' ',0,' ',0,10.0,5.0)
      CALL FRAME
      CALL GRAF(0.0,XSTEP,2.*MAXR,0.0,XSTEP,MAXR)

C-----------------------------------------------------------
C THE CONTOUR SEARCH SUBROUTINE IS CALLED WITH THE APPRO-
C PRIATE QUANTITY TO BE PLOTTED.
C-----------------------------------------------------------

      IF (POPT.EQ.1) THEN
         CALL CONTOR(PBAR,FOUND)
      END IF
      IF (POPT.EQ.2) THEN
         CALL CONTOR(PREFL,FOUND)
      END IF
      IF (POPT.EQ.3) THEN
         CALL CONTOR(TSUBA,FOUND)
      END IF
      IF (POPT.EQ.4) THEN
         CALL CONTOR(TAU,FOUND)
      END IF

C---------------------------------------------------
C THE APPROPRIATE HEADING IS PLACED ON THE PLOT.
C---------------------------------------------------

      IF (POPT.EQ.1) THEN
         NCHAR=44
         NAME=NAME1
      END IF
      IF (POPT.EQ.2) THEN
         NCHAR=45
         NAME=NAME2
      END IF
      IF (POPT.EQ.3) THEN
         NCHAR=44
         NAME=NAME3
```

```
      END IF
      IF (POPT.EQ.4) THEN
         NCHAR=42
         NAME=NAME4
      END IF

      CALL HEIGHT(0.22)
      CALL MESSAG(%REF(NAME),NCHAR,.75,7.2)
      CALL HEIGHT(0.19)

C-----------------------------------------------------------
C THE REMAINING PLOT SPECIFICATIONS ARE WRITTEN ON THE PLOT.
C-----------------------------------------------------------

      CALL MESSAG(%REF(GUN),32,4.5,6.8)
      CALL MESSAG(%REF(LABEL2),25,4.5,6.5)
      CALL REALNO(%REF(ELEANG),2,8.6,6.5)
      CALL MESSAG(%REF(LABEL3),26,4.5,6.2)
      CALL REALNO(%REF(ZHEIGH),2,8.6,6.2)
      CALL MESSAG('ASUBM(m/s)=',11,0.0,6.8)
      CALL REALNO(%REF(ASUBM),1,2.0,6.8)
      CALL MESSAG('PRATIO=',7,0.0,6.5)
      CALL REALNO(%REF(PRATIO),1,2.0,6.5)
      CALL MESSAG('VSUBP(m/s)=',11,0.0,6.2)
      CALL REALNO(%REF(VSUBP),1,2.0,6.2)
      CALL MESSAG('GAMMA=',6,0.0,5.9)
      CALL REALNO(%REF(GAMMP),2,2.0,5.9)
      CALL MESSAG('MU=   ',6,0.0,5.6)
      CALL REALNO(%REF(MU),2,2.0,5.6)
      IF (INTBAL) THEN
         CALL MESSAG('PROPELLANT MASS(kg)=',20,4.5,5.9)
         CALL REALNO(%REF(PROPM),5,7.7,5.9)
         CALL MESSAG('PROJECTILE MASS(kg)=',20,4.5,5.6)
         CALL REALNO(%REF(PROJM),4,7.7,5.6)
         CALL MESSAG('SPECIFIC FORCE(       )=',24,0.0,5.2)
         CALL MIXALF('INSTRUCTION')
         CALL MESSAG('m(EH.6)2(EXHX)/s(EH.6)2(EXHX)$',100,
     &   2.00,5.2)
         CALL RESET('MIXALF')
         CALL REALNO(%REF(SPECF),104,3.20,5.2)
      END IF
      CALL MESSAG(%REF(LABEL1),38,2.25,-.75)
      CALL ENDGR(0)
      CALL HEIGHT(.25)
      IF (NOGRID.EQ.0) THEN
         CALL YNONUM
         CALL TITLE(' $',-100,' ',1,' ',1,10.0,5.0)
         CALL POLAR(DTOR,XSTEP,5.0,0.0)
         GO TO 4122
      END IF
      CALL TITLE(' $',-100,' ',1,' ',0,10.0,5.0)

```

```
C-----------------------------------------------------------
C THE ACTUAL CONTOUR LINES ARE DRAWN IN THE LINEAR MODE.
C NOW, THE POLAR MODE IS CALLED IN ORDER TO SUPERIMPOSE
C THE DISTANCE SCALE AND ADD THE GRID IF ACTIVATED.
C-----------------------------------------------------------

      CALL POLAR(DTOR,XSTEP,5.0,0.0)
      RATIO(1)=1.
      RATIO(2)=1.
      CALL MRSCOD(.3,2,RATIO)
      CALL GRID(-30,1)
4122  CALL ENDPL(0)

      IF (TRMNUM.EQ.0) CALL XRETRO (480)
      IF (TRMNUM.EQ.3) PRINT*,ESC//'*dA'
      IF (TRMNUM .EQ. 4) THEN
         CALL XRETRO(480)
         PRINT*,ESC//'2'
      END IF
      CALL DONEPL

C-------------------------------------------------------------
C TO ERASE THE PLOT AND RETURN TO PROGRAM EXEUTION, THE USER
C MUST PRESS 'RETURN'.
C-------------------------------------------------------------

      WRITE(6,*)
      CALL SCLEAR(TRMNUM)
      IF (POPT.EQ.1.AND.KLAP.EQ.1) THEN
         WRITE(6,*)
         WRITE(6,*)'The scaling relations used in this program were'
         WRITE(6,*)'derived using pressure data of no more than'
         WRITE(6,*)'2.4 atm. However, this graph exceeds this pressure'
         WRITE(6,*)'and these higher pressure predictions may be'
         WRITE(6,*)'considered less accurate.'
         WRITE(6,*)
         WRITE(6,*)'Press return to continue.'
         READ(5,*) DUMMY
      END IF
      IF (POPT.EQ.3.AND.KLAP.EQ.1) THEN
         WRITE(6,*)
         WRITE(6,*)'Some of the contours exceed the data range used'
         WRITE(6,*)'in deriving the scaling relations.  Therefore,'
         WRITE(6,*)'The lower time values may be less accurate'
         WRITE(6,*)
         WRITE(6,*)'Press return to continue.'
         READ(5,'(A1)') DUMMY
      END IF

      GOTO 4000

4200  CALL SCLEAR(TRMNUM)
```

```
C-----------------------------
C THE PLOT SIZE CAN BE ALTRERED.
C-----------------------------

      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'Input factor for changing graph size.'
      WRITE(6,*)'Enter 1.0 to obtain the default size.'
      WRITE(6,*)
      READ *,XYFAC
      GOTO 4000

4300  CALL SCLEAR(TRMNUM)

C-------------------------------------------------------------------
C THE USER CAN ELIMINATE OR REACTIVATE THE POLAR GRID THAT IS
C SUPERIMPOSED ON THE PLOT.
C-------------------------------------------------------------------

      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'A polar grid is by default superimposed on the'
      WRITE(6,*)'contour maps. To erase this grid, enter OFF.'
      WRITE(6,*)
      WRITE(6,*)'If the grid is already deactivated, enter ON.'
      WRITE(6,*)'Hit return for no change.'
      WRITE(6,*)
      READ(5,'(A3)') ERASE
      IF (ERASE.EQ.'OFF') NOGRID=0
      IF (ERASE.EQ.'ON ') NOGRID=1
      GO TO 4000

C------------------------------------------------------------------
C  SECTION 5000 WRITES THE INPUT INFORMATION TO A FILE FOR
C  FUTURE USE.
C------------------------------------------------------------------

5000  CALL SCLEAR(TRMNUM)
      WRITE(6,*) 'Enter the name you wish to give the file.'
      READ(*,'(A15)') WFILE

      OPEN (UNIT=1,FILE=WFILE,STATUS='NEW')
      WRITE(1,'(A32)') GUN
      WRITE(1,*) D
      WRITE(1,'(A1)') ANS1
      IF (ANS1 .EQ. 'Y') THEN
         WRITE(1,*) PROPM
         WRITE(1,*) TRAVEL
         WRITE(1,*) CHAMV
         WRITE(1,*) PROJ
         WRITE(1,*) PROPB
```

```
            WRITE(1,'(A1)') WUNITS
            WRITE(1,*) SPECF
            WRITE(1,*) TEMPK
         ELSE
            WRITE(1,*) ASUBM
            WRITE(1,*) PRATIO
         END IF
         WRITE(1,*) AMBSS
         WRITE(1,*) VSUBP
         WRITE(1,*) GAMMP
         WRITE(1,*) MU
         WRITE(1,*) ELEANG
         WRITE(1,*) ZHEIGH
         CLOSE(UNIT=1)

         GOTO 105

 6000    CONTINUE

         STOP
         END
```

```

C-----------------------------------------------------------------------
C SUBROUTINE CONTOR SEARCHES FOR THE BEGINNING OF CONTOURS BY COM-
C PARING THE VALUE OF THE FUNCTION AT ADJACENT GRID SQUARES.  IT IS
C CALLED BY THE MAIN PROGRAM ONCE THE FUNCTIONAL VALUES AT ALL THE
C GRIDPOINTS HAVE BEEN DETERMINED.
C-----------------------------------------------------------------------

          SUBROUTINE CONTOR(GRIDPT,FOUND)
          PARAMETER(NL=40,Q=50,P=100)
          INTEGER XM(P),YM(Q),UNUSED(P,Q),FOUND(NL),OPEN
          INTEGER POPT,PP,QQ
          REAL H(NL),GRIDPT(P,Q),PBAR(P,Q),MAXR
          COMMON/GRID/NLEVLS,H,K,OPEN,XM,YM,UNION
          COMMON/INC/RINC,MAXR
          COMMON/PRESS/PBAR,POPT,KLAP
          COMMON WORK(12000),XPLOT(250),YPLOT(250),
        $ LSAV,KSAV,NOSTOR(NL),MSAV

          HFACT = 1000.0

          PP = P
          QQ = Q

          DO 410 I=1,P
  410     XM(I)=I
          DO 420 I=1,Q
```

```
 420   YM(I)=I
       JM=Q-1
       IM=P-1

C---------------------------------------------------------------
C ALL CONTOURS AT EACH LEVEL ARE DRAWN BEFORE MOVING ON TO THE NEXT
C LEVEL.
C---------------------------------------------------------------
C-----------------------------------------------------
C THE 'XPLOT' AND 'YPLOT' VALUES WHICH HAVE BEEN STORED IN
C THEIR RESPECTIVE ARRAYS DURING SUBROUTINE PLOT ARE FED
C INTO DISSPLA'S CONTOUR SETUP OPTION 'CONCRV', ALONG WITH
C THE APPROPRIATE PRESSURE LEVEL.
C-----------------------------------------------------
       CALL BCOMON(12000)
       CALL CONBGN
       MSAV = 1
       DO 430 K=1,NLEVLS
       KSAV=K
       LSAV=0

C-----------------------------------------------------------------
C THE ARRAY UNUSED IS INITIALIZED FOR THE CONTOUR LEVEL.  UNUSED(I,J)=
C 1 MEANS THAT THE I,J GRIDPOINT WILL BE USED IN DRAWING THE CONTOURS
C AT THIS LEVEL.  UNUSED(I,J)=0 MEANS THAT THE I,J GRIDPOINT IS NOT
C SIGNIFICANT.  AT EACH CONTOUR LEVEL, THE BOUNDARY OF THE GRID IS
C SCANNED FOR A POINT WHERE A CONTOUR CROSSES INTO THE GRID WITH THE
C HIGHER FUNCTIONAL VALUES ON THE RIGHT.  THIS CONDITION PREVENTS THE
C PROGRAM FROM RELOCATING CONTOURS IT HAS ALREADY DRAWN.  THE SCAN OF
C THE GRID BOUNDARY BEGINS IN THE LOWER LEFT CORNER AND PROCEEDS COUN-
C TERCLOCKWISE.
C
C IF AN INTERSECTION WITH THE GRID BOUNDARY IS FOUND, SUBROUTINE FOL-
C LOW IS CALLED TO FOLLOW THIS CONTOUR THROUGH THE GRID UNTIL IT
C EXITS FROM THE GRID.
C-----------------------------------------------------------------

         DO 440 J=2,JM
            DO 445 I=2,IM
               UNUSED(I,J)=0
               IF (GRIDPT(I-1,J).LT.H(K).AND.GRIDPT(I,J).GE.H(K))
     &         UNUSED(I,J)=1

 445        CONTINUE
 440     CONTINUE
       OPEN=1

C---------------------------------------
C FIRST, THE BOTTOM EDGE OF THE GRID IS SCANNED.
C---------------------------------------

         DO 450 I=2,P
```

```
            IF (GRIDPT(I-1,1).LT.H(K).AND.GRIDPT(I,1).GE.H(K)) THEN
               FOUND(K)=1
               CALL FOLLOW(GRIDPT,I,1,-1,0,UNUSED)
               CALL CONCRV(XPLOT(1),YPLOT(1),LSAV,HFACT*H(K))
               LSAV = 0
            END IF
 450     CONTINUE

C------------------------------
C THE RIGHT EDGE IS SCANNED.
C------------------------------

         DO 460 J=2,Q
            IF (GRIDPT(P,J-1).LT.H(K).AND.GRIDPT(P,J).GE.H(K)) THEN
               FOUND(K)=1
               CALL FOLLOW(GRIDPT,PP,J,0,-1,UNUSED)
               CALL CONCRV(XPLOT(1),YPLOT(1),LSAV,HFACT*H(K))
               LSAV = 0
            END IF
 460     CONTINUE

C----------------------------
C THE TOP EDGE IS SCANNED.
C----------------------------

         DO 470 L=1,IM
            I=P-L
            IF (GRIDPT(I+1,Q).LT.H(K).AND.GRIDPT(I,Q).GE.H(K)) THEN
                FOUND(K)=1
                CALL FOLLOW(GRIDPT,I,QQ,1,0,UNUSED)
                CALL CONCRV(XPLOT(1),YPLOT(1),LSAV,HFACT*H(K))
                LSAV = 0
            END IF
 470     CONTINUE

C------------------------------
C THE LEFT EDGE IS SCANNED.
C------------------------------

         DO 480 L=1,JM
            J=Q-L
            IF (GRIDPT(1,J+1).LT.H(K).AND.GRIDPT(1,J).GE.H(K)) THEN
               FOUND(K)=1
               CALL FOLLOW(GRIDPT,1,J,0,1,UNUSED)
               CALL CONCRV(XPLOT(1),YPLOT(1),LSAV,HFACT*H(K))
               LSAV = 0
            END IF
 480     CONTINUE

C-----------------------------------------------------------------------
C ONCE ALL THE OPEN CONTOURS AT LEVEL H ARE FOUND AND FOLLOWED TO
C COMPLETION, THE ARRAY UNUSED IS SEARCHED FOR ANY CLOSED CONTOURS
```

```
C OF HEIGHT H.  IF A CLOSED CONTOUR IS FOUND, SUBROUTINE FOLLOW IS
C CALLED TO FOLLOW IT UNTIL IT RETURNS TO THE POINT WHERE IT WAS
C FOUND.
C-----------------------------------------------------------------
          MSAV = 2
          OPEN=0
          DO 490 L=2,JM
             J=Q-L+1
             DO 500 M=2,IM
                I=P-M+1
                IF (UNUSED(I,J).EQ.1) THEN
                   FOUND(K)=1
                   CALL FOLLOW(GRIDPT,I,J,-1,0,UNUSED)
                   CALL CONCRV(XPLOT(1),YPLOT(1),LSAV,HFACT*H(K))
                   LSAV = 0
                END IF
 500         CONTINUE
 490      CONTINUE
 430   CONTINUE
       CALL CONEND

C-----------------------------------------------------------
C NEXT, THE DISSPLA PARAMETERS ARE SET TO DRAW SOLID CONTOUR
C LINES AND LABEL THEM WHERE POSSIBLE. THE 'CONTUR' COMMAND
C ACTUALLY DRAWS THE LINES USING THE DATA STORED IN 'CONCRV'.
C-----------------------------------------------------------

       CALL HEIGHT(.16)
       CALL CONLIN(0,'SOLID','LABELS',1,4)
       CALL CONANG(90.)
       CALL CONMIN(4.5)
       CALL CONTUR(1,'LABELS','DRAW')
       RETURN
       END
```

```

C-----------------------------------------------------------------
C SUBROUTINE FOLLOW FOLLOWS A CONTOUR THROUGH THE GRID ONCE IT HAS
C BEEN FOUND BY CHECKING THE SIDES OF THE GRID SQUARES TO DETERMINE
C WHICH SIDE THE CONTOUR PASSES THROUGH.  IT FINDS THE POINT OF
C INTERSECTION USING LINEAR INTERPOLATION BETWEEN THE ADJACENT GRID-
C POINTS.  CONTROL DOES NOT RETURN TO SUBROUTINE CONTOR UNTIL
C THE CONTOUR EITHER EXITS FROM THE GRID OR RETURNS TO THE POINT
C IN THE GRID WHERE IT WAS FIRST FOUND.  WHEN SUBROUTINE FOLLOW
C DETERMINES HOW THE CONTOUR PASSES THROUGH A GRID SQUARE, SUBROU-
C TINE PLOT IS CALLED TO STORE THE X AND Y VALUES LATER USED FOR
C THE ACTUAL PLOTTING.
C-----------------------------------------------------------------

```

```
      SUBROUTINE FOLLOW(GRIDPT,IIG,JJG,IIA,JJA,UNUSED)
      PARAMETER(NL=40,Q=50,P=100)
      INTEGER IG,JG,IA,JA,OPEN,K,LAST,FIRST,TE,XM(P),YM(Q),
     & NLEVLS,UNUSED(P,Q)
      REAL GRIDPT(P,Q),H(NL),Z,ZA,ZB,ZC,X,Y,
     & RINC,T,MAXR
      COMMON/GRID/NLEVLS,H,K,OPEN,XM,YM,ISIGN
      COMMON/INC/RINC,MAXR
      COMMON WORK(12000),XPLOT(250),YPLOT(250),
     & LSAV,KSAV,NOSTOR(NL),MSAV

C-------------------------------------------------------------------
C THE PARAMETERS IG,JG,IA, AND JA ARE USED IN VARIOUS COMBINATIONS
C AS THE SUBSCRIPTS FOR THE ARRAYS XM AND YM.  THE VALUES OF THESE
C SUBSCRIPTS DETERMINE WHICH GRID POINTS ARE CONSIDERED IN ANY PARTI-
C CULAR LINEAR INTERPOLATION.
C-------------------------------------------------------------------

      IG=IIG
      JG=JJG
      IA=IIA
      JA=JJA

C-------------------------------------------------------------------
C FIRST=1 INDICATES THAT A NEW CONTOUR HAS JUST BEEN FOUND; FIRST=0
C MEANS THAT THE PLOTTING OF THE CONTOUR IS ALREADY IN PROGRESS.
C LAST=1 INDICATES THAT THE END OF A CONTOUR HAS BEEN REACHED; LAST=0
C MEANS THAT THE CONTOUR IS NOT YET FINISHED.
C-------------------------------------------------------------------

      FIRST=1
      LAST=0

C---------------------------------------------------------------------
C THE LOCATION OF THE POINT T WHERE THE CONTOUR PASSES BETWEEN THE
C POINTS (I,J) AND (I+IA,J+JA) IS CALCULATED USING LINEAR INTERPOLATION.
C---------------------------------------------------------------------

      Z=GRIDPT(IG,JG)
      ZA=GRIDPT(IG+IA,JG+JA)
  605 T=0.
      IF(Z.NE.ZA) T=(Z-H(K))/(Z-ZA)
      X=XM(IG)-T*(XM(IG)-XM(IG+IA))
      Y=YM(JG)-T*(YM(JG)-YM(JG+JA))

C-------------------------------------------------------------------
C TESTS ARE NOW PERFORMED TO DETERMINE IF T IS THE LAST POINT ON THE
C CONTOUR.
C-------------------------------------------------------------------

      IF (OPEN.EQ.1) GOTO 610
      IF (IA.EQ.-1.AND.UNUSED(IG,JG).EQ.0) LAST=1
```

```
      GOTO 650
610   IF (FIRST.EQ.1) GOTO 660
      IF (JA.EQ.0) GOTO 620
      GOTO 630
620   IF (JG.EQ.1.OR.JG.EQ.Q) LAST=1
630   IF (JA.NE.0) GOTO 640
      GOTO 650
640   IF (IG.EQ.1.OR.IG.EQ.P) LAST=1
650   IF (LAST.EQ.1) GOTO 660
      IF (IA.EQ.-1) UNUSED(IG,JG)=0

C--------------------------------------------------------
C THE COORDINATES OF T ARE OUTPUT TO SUBROUTINE PLOT.
C--------------------------------------------------------

660    LSAV=LSAV+1
       CALL PLOT(X,Y,FIRST)

C------------------------------------------------------------------
C COMPARISONS ARE NOW MADE TO DETERMINE WHICH OF THE CELL SIDES
C THE CONTOUR CROSSES NEXT.  THE VALUES OF Z, ZA, IG, JG, IA,
C AND JA ARE ADJUSTED BEFORE CONTINUING TO FIND A NEW POINT, T.
C------------------------------------------------------------------

      IF (LAST.EQ.1) THEN
         NOSTOR(KSAV)=LSAV
         RETURN
      END IF
      ZB=GRIDPT(IG+JA,JG-IA)
      IF (ZB.GE.H(K)) GOTO 670
      ZA=ZB
      TE=IA
      IA=JA
      JA=-TE
      GOTO 690
670   ZC=GRIDPT(IG+IA+JA,JG-IA+JA)
      IF (ZC.GE.H(K)) GOTO 680
      Z=ZB
      ZA=ZC
      IG=IG+JA
      JG=JG-IA
      GOTO 690
680   Z=ZC
      IG=IG+IA+JA
      JG=JG-IA+JA
      TE=JA
      JA=IA
      IA=-TE
690   FIRST=0
      GOTO 605
      END
```

```
C-------------------------------------------------------------------
C SUBROUTINE PLOT STORES THE X AND Y VALUES IN TWO 2 DIMENSIONAL
C ARRAYS CALLED XPLOT AND YPLOT. THESE VALUES WILL BE USED IN THE
C CALL TO CONTUR FOR THE ACTUAL PLOTTING.
C-------------------------------------------------------------------

      SUBROUTINE PLOT(X,Y,FIRST)
      PARAMETER(NL=40,Q=50,P=100)
      INTEGER FIRST
      REAL MAXR
      COMMON/INC/RINC,MAXR
      COMMON WORK(12000),XPLOT(250),YPLOT(250),
     & LSAV,KSAV,NOSTOR(NL),MSAV
      XPLOT(LSAV)=(RINC*X)
      YPLOT(LSAV)=(RINC*Y)
      RETURN
      END
```

```
C-------------------------------------------------------------
C SUBROUTINE FNDMAX FINDS THE MAXIMUM VALUE OF ANY OF THE
C FUNCTIONS FOR WHICH CONTOURS ARE PLOTTED.
C-------------------------------------------------------------

      SUBROUTINE FNDMAX(VALS1,MAXVAL,POPT)
      PARAMETER(NL=40,Q=50,P=100)
      INTEGER POPT
      REAL VALS1(P,Q),MAXVAL
      MAXVAL=0.
      J=1
      DO 710 I=1,P
         DO 720 J=1,Q
            IF (POPT.EQ.1.AND.I.GE.45.AND.I.LE.55
     &          .AND.J.LE.5) GOTO 720
            IF (VALS1(I,J).GT.MAXVAL) MAXVAL=VALS1(I,J)
 720     CONTINUE
 710  CONTINUE
      RETURN
      END
```

```
C-------------------------------------------------------------
C SUBROUTINE FNDMIN FINDS THE MINIMUM VALUE OF ANY OF THE
```


61

```
C FUNCTIONS FOR WHICH CONTOURS ARE PLOTTED.
C-----------------------------------------------------------------

      SUBROUTINE FNDMIN(VALS2,MINVAL,POPT)
      PARAMETER(NL=40,Q=50,P=100)
      INTEGER POPT
      REAL VALS2(P,Q),MINVAL
      MINVAL=100000.
      DO 810 I=1,P
         DO 820 J=1,Q
            IF(POPT.EQ.3.AND.I.GE.45.AND.I.LE.55.AND.J.LE.5)
     &        GO TO 820
            IF(POPT.EQ.4.AND.I.GE.45.AND.I.LE.55.AND.J.LE.5)
     &        GO TO 820
            IF (VALS2(I,J).LT.MINVAL)  MINVAL=VALS2(I,J)
 820     CONTINUE
 810  CONTINUE
      RETURN
      END
```

```

C-----------------------------------------------------------------------
C SUBROUTINE REFCAL IS CALLED BY THE MAIN PROGRAM TO DETERMINE
C THE REFLECTED OVERPRESSURE, PREREF AT EACH GRIDPOINT IN THE CONTOUR
C GRID.  FIRST A CHECK IS MADE TO SEE IF REGULAR REFLECTION IS
C POSSIBLE.  IF IT IS POSSIBLE, THE REGULAR REFLECTION CALCULATION
C PROCEEDS.  IF REGULAR REFLECTION DOES NOT OCCUR SUBROUTINE
C NONREG IS CALLED TO DETERMINE THE REFLECTED OVERPRESSURE.
C-----------------------------------------------------------------------

      SUBROUTINE REFCAL(PRESIN,PREREF,ALPHA1)
      REAL MACHN1,MACHN2,LOWBND,PSIANG(11),ANGLIN(11),SLOPES(11)
      REAL PSISLP(11)
      COMMON/REFL/GAMMA,TOL,UPBND,LOWBND,XI
      COMMON/TRIG/RTOD,DTOR,PI
      COMMON/LAGRAN/PSIANG,PSISLP,ANGLIN,SLOPES
      CALL PRELIM(ALPHA1,PRESIN,DELTA,MACHN1,MACHN2,DELMAX)
      IF (DELTA.GE.0.90*DELMAX) THEN
         CALL NONREG(DELMAX,PRESIN,ALPHA1,MACHN1,MACHN2,PREREF)
         RETURN
      END IF
      ALPMAX=FALMAX(MACHN2,GAMMA)
      CALL ITRATE(0,ALPHA2,DELTA,LOWBND,ALPMAX,MACHN2,PRESIN)
      QUOTNT=(2.*GAMMA*(MACHN2**2*(SIN(ALPHA2))**2-1.))/
     & (GAMMA+1.)+1.
      PREREF=QUOTNT*XI-1.
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C SUBROUTINE PRELIM PERFORMS THE INITIAL OBLIQUE SHOCK CALCULATIONS FOR
C THE REGULAR REFLECTION SOLUTION.  MN1 IS THE STREAMLINE VELOCITY IN
C FRONT OF THE INCIDENT SHOCK.  DELANG IS THE FLOWSTREAM DEFLECTION
C ANGLE.  MN2 IS THE STREAMLINE VELOCITY BEHIND THE INCIDENT SHOCK.
C DELMA IS THE MAXIMUM FLOWSTREAM DEFLECTION ANGLE FOR REGULAR REFLECTION.
C-----------------------------------------------------------------------

      SUBROUTINE PRELIM(ANGINC,PRESIN,DELANG,MN1,MN2,DELMA)
      REAL MN1,MN2,MN,LOWBND
      COMMON/REFL/GAMMA,TOL,UPBND,LOWBND,XI
      SINANG=SIN(ANGINC)
      MN1=(SQRT(((GAMMA+1.)/(2.*GAMMA))*PRESIN+1.))/
     & SINANG
      MN=MN1*SINANG
      DELANG=ATAN(1./(TAN(ANGINC)*(((GAMMA+1.)*
     & MN1**2*.5/(MN**2-1.))-1.)))
      MN2=(SQRT(((GAMMA-1.)*XI+(GAMMA+1.))/(2*GAMMA*XI)))
     & /SIN(ANGINC-DELANG)
      IF (MN2.LT.1.) MN2=1.
      DELMA=(0.7698*(SQRT(MN2**2-1.))**3)/((GAMMA+1.)*MN2**2)
      RETURN
      END
```

```
C----------------------------------------------------------------
C SUBROUTINE NONREG DETERMINES THE REFLECTED OVERPRESSURE WHEN
C REGULAR REFLECTION DOES NOT OCCUR.
C----------------------------------------------------------------

      SUBROUTINE NONREG(DELMAX,PRESIN,ALPHA1,MACHN1,MACHN2,
     & PREREF)
      REAL LOWBND,MACHN1,MACHN2,PSIANG(11),ANGLIN(11),SLOPES(11)
      REAL PSISLP(11)
      COMMON/TRIG/RTOD,DTOR,PI
      COMMON/REFL/GAMMA,TOL,UPBND,LOWBND,XI
      COMMON/LAGRAN/PSIANG,PSISLP,ANGLIN,SLOPES
      COMMON/ATMOS/ATM

C------------------------------------------------------------------
C FIRST, THE ANGLE OF INCIDENCE, ALPHD1, WHERE REGULAR REFLECTION
C STOPS IS DETERMINED BY CALLING SUBROUTINE ALMAX.  NEXT, THE
C ANGLE OF INCIDENCE, STTANG, WHERE THE LINEAR APPROXIMATION BEGINS
C IS DETERMINED BY SUBROUTINE LAGRNG.
C------------------------------------------------------------------

```

```
      CALL ALMAX(ALPHD1,PRESIN)
      CALL LAGRNG(ATM*PRESIN,STTANG,PSIANG,ANGLIN)
      ALPHD2=ALPHD1-.001

C-----------------------------------------------------------------
C THE FOLLOWING SECTION OF CODE PERTAINS TO THE CASE WHEN ALPHA1
C IS IN THE REGION WHERE THE CUBIC POLYNOMIAL IS USED TO APPROXI-
C MATE THE REFLECTED OVERPRESSURE.  THE SLOPE, SLOPE1, AT ALPHD1
C IS DETERMINED BY FINITE DIFFERENCE.  THE SLOPE, SLOPE2, AT STTANG
C IS DETERMINED BY SUBROUTINE LAGRNG.  SUBROUTINE CUBIC IS CALLED
C TO DETERMINE THE CUBIC EQUATION AND EVALUATE IT AT ALPHA1 TO OBTAIN
C THE REFLECTED OVERPRESSURE.
C-----------------------------------------------------------------

      IF (ALPHA1.GT.ALPHD2.AND.ALPHA1.LT.STTANG) THEN
         CALL PRELIM(ALPHD1,PRESIN,DELTA,MACHN1,MACHN2,DELMAX)
         ALPHM1=FALMAX(MACHN2,GAMMA)
         CALL ITRATE(0,ALPHP1,DELTA,LOWBND,ALPHM1,MACHN2,PRESIN)
         QUOTNT=(2.*GAMMA*(MACHN2**2*(SIN(ALPHP1))**2-1.))/
     &   (GAMMA+1.)+1.
         PREFD1=QUOTNT*XI-1.
         CALL PRELIM(ALPHD2,PRESIN,DELTA,MACHN1,MACHN2,DELMAX)
         ALPHM2=FALMAX(MACHN2,GAMMA)
         CALL ITRATE(0,ALPHP2,DELTA,LOWBND,ALPHM2,MACHN2,PRESIN)
         QUOTNT=(2.*GAMMA*(MACHN2**2*(SIN(ALPHP2))**2-1.))/
     &   (GAMMA+1.)+1.
         PREFD2=QUOTNT*XI-1.
         SLOPE1=(PREFD1-PREFD2)/(.001*PRESIN)
         CALL LAGRNG(ATM*PRESIN,SLOPE2,PSISLP,SLOPES)
         VALSTT=FLINE(PRESIN,STTANG)
         CALL CUBIC(ALPHD2,PREFD2/PRESIN,SLOPE1,STTANG,VALSTT,SLOPE2,
     &   ALPHA1,PREREF)
         PREREF=PREREF*PRESIN
      END IF

C-------------------------------------------------------------------
C IF ALPHA1>STTANG, THEN THE LINEAR APPROXIMATION IS USED TO DETERMINE
C THE REFLECTED OVERPRESSURE.
C-------------------------------------------------------------------

      IF (ALPHA1.GT.ALPHD1.AND.ALPHA1.GE.STTANG) PREREF=
     & FLINE(PRESIN,ALPHA1)*PRESIN
      RETURN
      END
```

```

C----------------------------------------------------------------
C SUBROUTINE ITRATE PERFORMS AN ITERATIVE PROCEDURE TO DETER-
C MINE THE SHOCK WAVE ANGLE OF INCIDENCE, ANGLE, FOR A GIVEN
```

```
C FLOW DEFLECTION ANGLE, ANGLE2.  THE FUNCTION WHICH IS ITER-
C ATED IS F.  A SOLUTION HAS BEEN FOUND WHEN F=0.
C-------------------------------------------------------------

      SUBROUTINE ITRATE(L,ANGLE,ANGLE2,ANGLOW,ANGLHI,
     & MN2,PRESIN)
      INTEGER SIGN,SIGNH,SIGNL
      REAL MN2,ANG(20),VAL(20),LOWBND
      COMMON/TRIG/RTOD,DTOR,PI
      COMMON/REFL/GAMMA,TOL,UPBND,LOWBND,XI
      ANGLEL=ANGLOW
      ANGLEH=ANGLHI

C-------------------------------------------------------------------
C THE RANGE OF VALUES FOR ANGLE1 IS DIVIDED INTO 20 EQUAL INTER-
C VALS AND F IS EVALUATED AT EACH INTERVAL.  THE SMALLEST VALUE OF
C F IS FOUND.  THE ARGUMENT ANGLE FOR THIS VALUE OF F BECOMES ANGLE
C AND THE ANGLES TO THE LEFT AND RIGHT OF THIS POINT BECOME ANGLEL
C AND ANGLEH RESPECTIVELY.
C-------------------------------------------------------------------

      ANGINC=(ANGLEH-ANGLEL)/19.
      DO 910 I=1,20
         ANG(I)=ANGLEL+(I-1)*ANGINC
         VAL(I)=F(L,ANG(I),ANGLE2,MN2,GAMMA,PRESIN)
  910 CONTINUE
      VALMIN=1000000.
      DO 920 I=1,20
         IF (ABS(VAL(I)).LT.VALMIN) THEN
            VALMIN=ABS(VAL(I))
            IMIN=I
         END IF
  920 CONTINUE
      ILEFT=IMIN-1
      IRIGHT=IMIN+1
      IF (IMIN.EQ.20) IRIGHT=IMIN
      IF (IMIN.EQ.1) ILEFT=IMIN
      ANGLEL=ANG(ILEFT)
      ANGLEH=ANG(IRIGHT)
      ANGLE=(ANGLEH-ANGLEL)/2.+ANGLEL

C-------------------------------------------------------------
C THE SIGNS OF ANGLEL, ANGLE, AND ANGLEH ARE COMPARED TO
C DETERMINE WHETHER THE ZERO OF F IS GREATER OR LESS THAN
C ANGLE.  SUPPOSE IT IS DETERMINED THAT THE ROOT IS LESS THAN
C ANGLE.  THE INTERVAL FROM ANGLEL TO ANGLE IS DIVIDED
C IN HALF AND ANGLE BECOMES ANGLEH AND THE MIDDLE POINT
C BECOMES ANGLE.  F IS EVALUATED AND THE SIGNS ARE AGAIN
C COMPARED.  THE PROCESS REPEATS UNTIL ABS[F(ANGLEL)-
C F(ANGLEH)]<TOL.
C-------------------------------------------------------------

```

```
 900  VALUEL=F(L,ANGLEL,ANGLE2,MN2,GAMMA,PRESIN)
      VALUEH=F(L,ANGLEH,ANGLE2,MN2,GAMMA,PRESIN)
      VALUE=F(L,ANGLE,ANGLE2,MN2,GAMMA,PRESIN)
      IF (VALUEH.LT.0.) SIGNH=-1
      IF (VALUEH.GE.0.) SIGNH=1
      IF (VALUEL.LT.0.) SIGNL=-1
      IF (VALUEL.GE.0.) SIGNL=1
      IF (VALUE.LT.0.)  SIGN=-1
      IF (VALUE.GE.0.)  SIGN=1
      IF (SIGN.EQ.SIGNL) THEN
         TEMP=ANGLE
         ANGLE=ANGLE+(ANGLEH-ANGLE)/2.
         ANGLEL=TEMP
      END IF
      IF (SIGN.EQ.SIGNH) THEN
         TEMP=ANGLE
         ANGLE=ANGLE-(ANGLE-ANGLEL)/2.
         ANGLEH=TEMP
      END IF
      DIF=ABS(VALUEH-VALUEL)
      IF (DIF.GT.TOL) GOTO 900
      RETURN
      END
```

```

C-----------------------------------------------------------------------
C SUBROUTINE ALMAX DTERMINES THE MAXIMUM ANGLE OF INCIDENCE, ALPHD1,
C FOR ANY INCIDENT OVERPRESSURE, PRESIN.
C-----------------------------------------------------------------------

      SUBROUTINE ALMAX(ALPHD1,PRESIN)
      REAL MACHN1,MACHN2
      COMMON/TRIG/RTOD,DTOR,PI
      DO 1010 I=30,90,5
         ALPHCH=DTOR*I
         CALL PRELIM(ALPHCH,PRESIN,DELTA,MACHN1,MACHN2,DELMAX)
         DELMAX=0.90*DELMAX
         IF (DELTA.GT.DELMAX) THEN
            DO 1020 J=I-5,I
               ALPHCH=DTOR*J
               CALL PRELIM(ALPHCH,PRESIN,DELTA,MACHN1,MACHN2,
     &                     DELMAX)
               DELMAX=0.90*DELMAX
               IF (DELTA.GT.DELMAX) THEN
                  DO 1030 K=1,10
                     ALPHCH=DTOR*((J-1)+.1*K)
                     CALL PRELIM (ALPHCH,PRESIN,DELTA,MACHN1,
     &                            MACHN2,DELMAX)
                     DELMAX=0.90*DELMAX
```

```
                     IF (DELTA.GT.DELMAX) THEN
                        ALPHD1=DTOR*((J-1)+.1*(K-1))
                        RETURN
                     END IF
 1030             CONTINUE
                END IF
 1020         CONTINUE
            END IF
 1010     CONTINUE
          END
```

```
C------------------------------------------------------------------
C SUBROUTINE CUBIC FORMS AND EVALUATES THE THIRD DEGREE POLYNOMIAL
C WHICH FITS THE MIDDLE SECTION OF THE REFLECTED OVERPRESSURE VS.
C ANGLE OF INCIDENCE CURVES.
C------------------------------------------------------------------

      SUBROUTINE CUBIC(A,D,E,F,I,J,X,Y)
      REAL I,J,K,L
      COMMON/REFL/GAMMA,TOL,UPBND,LOWBND,XI
      B=A**2/2.
      C=A**3/6.
      G=F**2/2.
      H=F**3/6.
      K=G-B
      L=A-F
      COEFF4=(L*(I-D+E*L)+(J-E)*(K+A*L))/(L*(H-C+B*L)+K*(K+A*L))
      COEFF3=((I-D+E*L)-(H-C+B*L)*COEFF4)/(K+A*L)
      COEFF2=E-A*COEFF3-B*COEFF4
      COEFF1=D-A*COEFF2-B*COEFF3-C*COEFF4
      Y=(X**3)*COEFF4/6.+(X**2)*COEFF3/2.+X*COEFF2+COEFF1
      RETURN
      END
```

```
C-------------------------------------------------------------------
C SUBROUTINE LAGRNG PERFORMS LAGRANGIAN INTERPOLATION TO DETER-
C MINE THE VALUE, Y, OF A FUNCTION AT ARGUMENT, X.  THE ELEMENTS
C OF ARRAY FUNCT ARE THE TABULATED FUNCTIONAL VALUES FOR THE COR-
C RESPONDING ARGUMENTS IN ARGUM.
C-------------------------------------------------------------------

      SUBROUTINE LAGRNG(X,Y,ARGUM,FUNCT)
      REAL ARGUM(11),FUNCT(11),DIFF(11),A(11)

```

```
C-----------------------------------------------------------------
C FIRST, THE FOUR VALUES IN ARRAY ARGUM NEAREST IN ABSOLUTE VALUE
C TO X ARE SELECTED ALONG WITH THE CORRESPONDING ELEMENTS OF FUNCT.
C THESE ARE THE OPTIMUM POINTS TO USE TO FORM A THIRD DEGREEE
C LAGRANGIAN POLYNOMIAL TO THE FUNCTION AT X.
C-----------------------------------------------------------------

      DO 1110 I=1,11
         DIFF(I)=APS(X-ARGUM(I))
 1110 CONTINUE
      SORTED=0
      LAST=11
      DO WHILE (SORTED.NE.1.AND.LAST.GE.2)
         SORTED=1
         LIMIT=LAST-1
         DO 1120 I=1,LIMIT
            IF (DIFF(I).GT.DIFF(I+1)) THEN
               SORTED=0
               TEMP1=DIFF(I)
               TEMP2=ARGUM(I)
               TEMP3=FUNCT(I)
               DIFF(I)=DIFF(I+1)
               ARGUM(I)=ARGUM(I+1)
               FUNCT(I)=FUNCT(I+1)
               DIFF(I+1)=TEMP1
               ARGUM(I+1)=TEMP2
               FUNCT(I+1)=TEMP3
            END IF
 1120    CONTINUE
         LAST=LAST-1
      END DO
      Y=0.
 1125 CONTINUE

C-----------------------------------------------------------------
C THE THIRD DEGREE POLYNOMIAL IS FORMED AND EVALUATED AT X TO
C DETERMINE Y.
C-----------------------------------------------------------------

      DO 1130 K=1,4
         A(K)=1.
         DO 1140 J=1,4
            IF (J.NE.K) THEN
               A(K)=A(K)*(X-ARGUM(J))/(ARGUM(K)-ARGUM(J))
            END IF
 1140    CONTINUE
         Y=Y+A(K)*FUNCT(K)
 1130 CONTINUE
      RETURN
      END
```

```
C----------------------------------------------------------------
C FUNCTION FLINE FORMS AND EVALUATES THE LINEAR FIT SECTION OF
C THE REFLECTED OVERPRESSURE CURVES.  PRESIN IS THE INCIDENT OVER-
C PRESSURE AND ANGINC IS THE INCIDENT WAVE ANGLE AT WHICH THE
C LINEAR EQUATION IS EVALUATED.
C----------------------------------------------------------------

      REAL FUNCTION FLINE(PRESIN,ANGINC)
      REAL M,PSIANG(11),PSISLP(11),ANGLIN(11),SLOPES(11)
      COMMON/TRIG/RTOD,DTOR,PI
      COMMON/LAGRAN/PSIANG,PSISLP,ANGLIN,SLOPES
      COMMON/ATMOS/ATM

C-----------------------------------------------------------
C SUBROUTINE LAGRNG IS CALLED TO DETERMINE THE SLOPE, M
C-----------------------------------------------------------

      CALL LAGRNG(ATM*PRESIN,M,PSISLP,SLOPES)
      FLINE=M*ANGINC-M*PI/2.+1.
      RETURN
      END
```

```
C--------------------------------------------------------------
C SUBROUTINE SCLEAR ERASES THE SCREEN AND SENDS THE CURSOR HOME
C IF USING A RETRO VT, HP2623, OR TEKTRONIKS TERMINAL.
C--------------------------------------------------------------

      SUBROUTINE SCLEAR(TRMNUM)
      INTEGER TRMNUM
      INTEGER*4 CHANNEL,STAT
      CHARACTER ESC/27/
      COMMON/IO/CHANNEL
         IF(TRMNUM .EQ. 0 .OR. TRMNUM .EQ. 4) THEN
            IAA=LIB$ERASE_PAGE(1,1)
            IDA=SCR$SET_CURSOR(%VAL(1),%VAL(1))
         ELSE IF(TRMNUM.EQ.3) THEN
            CALL ASSIGN_CHANNEL
            CALL OUTPUT(ESC//'H'//ESC//'J')
            CALL DEASSIGN_CHANNEL
         ELSE IF (TRMNUM.EQ.2) THEN
            CALL ASSIGN_CHANNEL
            CALL OUTPUT(ESC//CHAR(12))
```

```
            CALL DEASSIGN_CHANNEL
            DO 1555 I=1,50
               DO 1550 J=1,10000
 1550          CONTINUE
 1555       CONTINUE
         END IF
      RETURN
      END
```

```

C------------------------------------------------------------------
C  THE FOLLOWING SUBROUTINES ARE USED TO CLEAR THE SCREEN ON THE
C  HP2623 AND TEKTRONIKS TERMINALS USING SYSTEM SERVICE COMMANDS.
C------------------------------------------------------------------

        SUBROUTINE INPUT(TYPEIN)
        INCLUDE '($IODEF)'
        CHARACTER*(*) TYPEIN
        INTEGER*4 QUAD(2)
        INTEGER*4 CHANNEL,STAT,SYS$QIOW
        DATA QUAD/2*0/

        STAT=SYS$QIOW(,$VAL(CHANNEL),$VAL(IO$_TTYREADALL.OR.IO$M_NOECHO)
     +             ,,,,$REF(TYPEIN),$VAL(LEN(TYPEIN)),,QUAD,,,,)
        IF (STAT) RETURN
            PRINT*, 'READ ERROR ',STAT
            CALL DEASSIGN_CHANNEL
            STOP
        END
```

```

C-----------------
C SUBROUTINE OUTPUT
C-----------------

        SUBROUTINE OUTPUT(TYPEOUT)
        COMMON/IO/CHANNEL
        INCLUDE '($IODEF)'
        CHARACTER*(*) TYPEOUT
        INTEGER*4 CHANNEL,STAT,SYS$QIOW

        STAT=SYS$QIOW(,$VAL(CHANNEL),$VAL(IO$_WRITELBLK.OR.IO$M_NOFORMAT)
     +              ,,,,$REF(TYPEOUT),$VAL(LEN(TYPEOUT)),,$VAL(0),,,)
        IF (STAT) RETURN
           PRINT*,'WRITE ERROR',STAT
           CALL DEASSIGN_CHANNEL
```

```
         STOP
      END
```

```
C------------------
C SUBROUTINE ASSIGN
C------------------

      SUBROUTINE ASSIGN_CHANNEL
      COMMON/IO/CHANNEL
      INTEGER*4 SYS$ASSIGN,CHANNEL,STAT
      STAT=SYS$ASSIGN('TT',CHANNEL,,)
      IF (STAT) RETURN
         PRINT*, 'DEVICE ASSIGNMENT ERROR ',STAT
         STOP
      END
```

```
C--------------------
C SUBROUTINE DEASSIGN
C--------------------

      SUBROUTINE DEASSIGN_CHANNEL
      COMMON/IO/CHANNEL
      INCLUDE'($IODEF)'
      INTEGER*4 SYS$DASSGN,CHANNEL,STAT

      STAT=SYS$DASSGN(%VAL(CHANNEL))
      IF (.NOT.STAT) PRINT*, 'DEVICE DEASSIGN ERROR ',STAT
      RETURN
      END
```

```
C------------------------------------------------------------------
C FUNCTION FALMAX DETERMINES THE REFLECTED SHOCK WAVE ANGLE FOR
C MAXIMUM STREAM DEFLECTION BEHIND THE REFLECTED SHOCK.
C------------------------------------------------------------------

      FUNCTION FALMAX(MACHN2,GAMMA)
      REAL MACHN2
      SINSQA=(1./(4.*GAMMA*MACHN2**2))*((GAMMA+1.)*MACHN2**2-
     & 3.+GAMMA+SQRT((GAMMA+1.)*((GAMMA+1.)*MACHN2**4-
     & 2.*(3.-GAMMA)*MACHN2**2+(GAMMA+9.))))
```

```
      FALMAX=ASIN(SQRT(SINSQA))
      RETURN
      END
```

```

C-----------------------------------------------------------------------
C FUNCTION F IS THE FUNCTION EVALUATED IN SUBROUTINE ITRATE TO
C DETERMINE THE REFLECTED SHOCK WAVE ANGLE FOR A GIVEN FLOW DEFLEC-
C TION ANGLE.
C-----------------------------------------------------------------------

      FUNCTION F(L,ARGUM,KNOWN,VEL,GAMMA,PRESS)
      REAL KNOWN
      VELTEM=VEL
      IF (L.EQ.1) VELTEM=(SQRT(((GAMMA+1.)/(2.*GAMMA))*
     &                     PRESS+1.))/SIN(ARGUM)
      F=2.*(VELTEM**2*(SIN(ARGUM))**2-1.)/(TAN(ARGUM)*
     & (VELTEM**2*(GAMMA+COS(2.*ARGUM))+2.))-TAN(KNOWN)
      RETURN
      END
```

```

C-----------------------------------------------------------------------
C ONESIG CONVERTS X TO ONE SIGNIFICANT FIGURE.
C-----------------------------------------------------------------------

      FUNCTION ONESIG(X)

      IF ( X .GT. 9. ) THEN
         DO 10 I=1,10
            X = X/10.
            IF (X .LT. 9.) THEN
               X = ANINT(X)*(10.**I)
               ONESIG = X
               RETURN
            END IF
 10      CONTINUE
      END IF
      IF ( X .GE. .9 .AND. X .LE. 9. ) THEN
         ONESIG = ANINT(X)
         RETURN
      END IF
      IF ( X .LT. .9 ) THEN
         DO 20 I=1,10
            X = X*10.
            IF ( X .GT. .9 ) THEN
```

```
                X = ANINT(X)/(10.**I)
                ONESIG = X
                RETURN
             END IF
   20     CONTINUE
       END IF

       END
```

```

C-----------------------------------------------------------------------
C  TWOSIG CONVERTS X TO TWO SIGNIFICANT FIGURES.
C-----------------------------------------------------------------------

       FUNCTION TWOSIG(X)

       IF ( X .GT. 99.0 ) THEN
          DO 10 I=1,10
             X = X/10.
             IF (X .LT. 99.0) THEN
                X = ANINT(X)*(10.**I)
                TWOSIG = X
                RETURN
             END IF
   10     CONTINUE
       END IF
       IF ( X .GE. 9.9 .AND. X .LE. 99.0 ) THEN
          TWOSIG = ANINT(X)
          RETURN
       END IF
       IF ( X .LT. 9.9 ) THEN
          DO 20 I=1,10
             X = X*10.
             IF ( X .GT. 9.9 ) THEN
                X = ANINT(X)/(10.**I)
                TWOSIG = X
                RETURN
             END IF
   20     CONTINUE
       END IF

       END
```

LIST OF SYMBOLS

A_e	area of bore
a_m	propellant sound speed at muzzle immediately before projectile ejection
a_∞	ambient sound speed
B	fraction of propellant burnt
C	propellant mass
D	bore diameter of gun
E	internal energy of propellant gas immediately prior to projectile ejection
h	distance from the origin of the contour plane to the muzzle
ℓ	scale length for explosion
ℓ'	effective scaling length that varies with angle from boreline
m_p	projectile mass
m_1	effective projectile mass accounting for bore friction ($= 1.05\ m_p$)
M	Mach number of incident shock
M_1	streamline Mach number in front of incident shock
M_2	streamline Mach number behind incident shock
$\vec{n}$	unit vector normal to contour plane
$\vec{p}$	vector directed from the boreline to the field point of interest that is normal to the shockwave surface
p_m	muzzle pressure for propellant immediately before projectile ejection
p_∞	ambient pressure
pI	pressure behind incident shock
p_R	pressure behind reflected shock
$\overline{P}$	incident overpressure (atm)
$\overline{P}_R$	reflected overpressure (atm)
$\vec{r}$	vector directed from muzzle to field point of interest

r	magnitude of $\vec{r}$
R	gas constant
t_a	blast wave time of arrival
T_m	propellant temperature at muzzle immediately before uncorking
T_a	adiabatic flame temperature of propellant
T_{mean}	mean temperature of propellant gas
$\vec{u}$	unit vector parallel to boreline
U	combined volume of chamber and bore
V_p	exit velocity of projectile
x	coordinate having origin at the muzzle with the direction defined as perpendicular to the boreline and parallel to the contour plane. Shown in Figure 1.
y	coordinate having origin at the muzzle with the direction parallel to the contour plane and being in the vertical plane encompassing the boreline. Shown in Figure 1.
z	coordinate having origin at the muzzle with direction perpendicular to the contour plane. Positive direction corresponds to increasing distance from contour plane as shown in Figure 1.
Z	$(r/\ell')^{1.1}$
α_1	wave angle of incident shock
α_2	wave angle of reflected shock
γ	specific heat ratio
δ_1	flow deflection angle through incident shock
δ_2	flow deflection angle through reflected shock
δ_{max}	maximum stream deflection consistent with regular shock reflection
η	angle between $\vec{r}$ and $\overline{P}$
θ	polar angle from boreline to field point
μ	momentum index
$\vec{\xi}$	vector along boreline designating apparent origin of blast wave

LIST OF SYMBOLS (CONTINUED)

π	cubic polynomial used in the approximation of the reflected pressure coefficient curves
τ	blast wave positive phase duration
ϕ	angle between boreline and contour plane
χ	ratio of heat losses to kinetic energy
Ω	roughness factor

DISTRIBUTION LIST

No. of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
1	Commander US Army Materiel Command ATTN: AMCLDC 5001 Eisenhower Avenue Alexandria, VA 22333
2	Commander US Army Aviation Research and Development Command ATTN: Tech Dir AMSAV-E 4300 Goodfellow Boulevard St. Louis, MO 63120
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703-5301
1	President US Army Aviation Test Board ATTN: ATZQ-OP-AA Ft. Rucker, AL 36360
1	Commander US Army Communications - Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703

No. of Copies	Organization
1	Commander US Army Missile Command ATTN: AMSMI-R Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-RBL Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-TLH Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-RDK Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-YDL Redstone Arsenal, AL 35898
5	Commander US Army Tank Automotive Command ATTN: AMSTA-TSL PM-TANK PM-M1A1 PM-M1 PM-M60 Warren, MI 48090
1	Commander US Army Operational Test and Evaluation Agency ATTN: CSTE-ZS 5600 Columbia Pike Falls Church, VA 22041
1	Commander US Army Armament, Munitions & Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299
1	HQDA DAMA-ART-M Washington, DC 20310

DISTRIBUTION LIST (continued)

No. of Copies	Organization
7	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TSS SMCAR-TDS, Mr. Lindner SMCAR-LC-F, Mr. Loeb SMCAR-LCW, Mr. Salsbury SMCAR-LCW, Mr. R. Wrenn SMCAR-SCA, Mr. Gehbauer SMCAR-LCU, Mr. Barrieres Dover, NJ 07801-5001
1	ODCSI, USAREUR / 7A ATTN: AEAGB-PDN(S&E) APO, NY 09403
4	Director Benet Weapons Laboratory Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCB-TL Dr. T. Simkins Dr. G. Carofano Dr. C. Andrade Watervliet, NY 12189
1	Commander US Army Jefferson Proving Ground Madison, IN 47251
1	Commander US Army Materials and Mechanics Research Center ATTN: AMXRE-ATL Watertown, MA 02172
1	Commander US Army Natick Research and Development Laboratory ATTN: AMDNA, Dr. D. Sieling Natick, MA 01760
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missile Range, NM 88002
1	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Ft. Benning, GA 31905
1	Commander US Army Research Office Durham ATTN: CRD-AA-EH P.O. Box 12211 Research Triangle Park, NC 27709-2211
1	Commander US Army Ballistic Missile Defense Systems Command P.O. Box 1500 Huntsville, AL 35807
1	President US Army Armor and Engineer Board ATTN: ATZK-AE-PD, Mr. W. Wells Ft. Knox, KY 40121
1	Commander Naval Air Systems Command ATTN: AIR-604 Washington, DC 20360
1	Commander Naval Sea Systems Cmd ATTN: 003 Washington, DC 20362
1	Commander David W. Taylor Naval Ship Research & Development Ctr ATTN: Lib Div, Code 522 Aerodynamic Lab Bethesda, MD 20084
2	Commander Naval Surface Weapons Center ATTN: 6X Dr. J. Yagla Dr. G. Moore Dahlgren, VA 22448
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TDC Dover, NJ 07801

No. of Copies	Organization
1	Commander Naval Surface Weapons Center ATTN: Code 730 Silver Spring, MD 20910
1	Commander Naval Weapons Center ATTN: Technical Services Branch China Lake, CA 93555
1	Commander Naval Weapons Center ATTN: Tech Info Div Washington, DC 20375
1	Commander Naval Ordnance Station ATTN: Code FS13A, P. Sewell Indian Head, MD 20640
2	AFATL (DLRA) ATTN: F. Burgess Tech Lib Eglin AFB, FL 32542
1	AFWL/SUL Kirtland AFB, NM 87117
1	ASD/XRA (STINFO) Wright-Patterson AFB, OH 45433
1	Air Force Armament Laboratory ATTN: AFATL/DLODL Eglin AFB, FL 32542-5000
1	Commander US Army Development and Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433
1	Director Jet Propulsion Laboratory ATTN: Tech Lib 4800 Oak Grove Drive Pasadena, CA 91109
1	Director National Aeronautics and Space Administration George C. Marshall Space Flight Center ATTN: MS-I, Lib Huntsville, AL 38512

No. of Copies	Organization
1	Director NASA Scientific & Technical Information Facility ATTN: SAK/DL P.O. Box 8757 Baltimore/Washington International Airport, MD 21240
1	AAI Corporation ATTN: Dr. T. Stastny P.O. Box 126 Cockeysville, MD 21030
1	Advanced Technology Labs ATTN: Mr. J. Erdos Merrick & Steward Avenue Westbury, NY 11590
1	Aerospace Corporation ATTN: Dr. G. Widhopf Bldg. D8 M4/965 P.O. Box 92957 Los Angeles, CA 90009
1	AVCO Systems Division ATTN: Dr. D. Siegelman 201 Lowell Street Wilmington, MA 01887
1	Technical Director Colt Firearms Corporation 150 Huyshope Avenue Hartford, CT 14061
1	General Electric Armament & Electric Systems ATTN: Mr. R. Whyte Lakeside Avenue Burlington, VT 05401
2	Honeywell, Inc. ATTN: Mail Station MN 112190, G. Stilley MN 50-2060, Mr. T. Melander 600 Second Street, North East Hopkins, MN 55343

No. of Copies	Organization
1	Hughes Helicopter Company Bldg. 2, MST22B ATTN: Mr. R. Forker Centinella and Teale Streets Culver City, CA 90230
1	Martin Marietta Aerospace ATTN: Mr. A. J. Culotta P.O. Box 5837 Orlando, FL 32805
1	AEROJET Ordnance Company ATTN: Mr. A. Flatau 2521 Michelle Drive Tustin, CA 92680
1	Olin Corporation 275 Winchester Avenue New Haven, CT 06504
1	Director Sandia National Laboratory ATTN: Aerodynamics Dept Org 5620, R. Maydew Albuquerque, NM 87115
1	Guggenheim Aeronautical Lab California Institute of Tech ATTN: Tech Lib Pasadena, CA 91104
1	Franklin Institute ATTN: Tech Lib Race & 20th Streets Philadelphia, PA 19103
1	Director Applied Physics Laboratory The Johns Hopkins University Johns Hopkins Road Laurel, MD 20707
1	Massachusetts Institute of Technology Dept of Aeronautics and Astronautics ATTN: Tech Lib 77 Massachusetts Avenue Cambridge, MA 02139

No. of Copies	Organization
1	Ohio State University Dept of Aeronautics and Astronautical Engineering ATTN: Tech Lib Columbus, OH 43210
1	Polytechnic Institute of New York Graduate Center ATTN: Tech Lib Route 110 Farmingdale, NY 11735
2	Loral Corporation ATTN: S. Schmotolocha Ben Axley 300 N. Halstead St. P.O. Box 7101 Pasadena, CA 91109
1	Director Forrestal Research Center Princeton University Princeton, NJ 08540
1	Kaman Tempo ATTN: Mr. J. Hindes 816 State Street P.O. Drawer QQ Santa Barbara, CA 93102
1	Southwest Research Institute ATTN: Mr. Peter S. Westine P.O. Drawer 28510 8500 Culebra Road San Antonio, TX 78228
1	Boeing Company ATTN: C. R. Pond MS 8C-64 PO Box 3999 Seattle, WA 98124
1	FMC Corporation Northern Ordnance Division ATTN: Scott Langlie, Advanced Techniques 4800 East River Road Minneapolis MN 55421

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: AMXSY-D
AMXSY-MP, H. Cohen

Cdr, USATECOM
ATTN: AMSTE-TO-F

Cdr, CRDC, AMCCOM
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

Dir, Wpns Sys Concepts Team
ATTN: AMSMC-ACW

Dir, USAHEL
ATTN: Dr. Weisz
Dr. Cummings
Mr. Garinther

Dir, USACSTA
ATTN: Mr. S.Walton